RESOLUTION ENHANCEMENT USING NATURAL IMAGE STATISTICS AND MULTIPLE ALIASED OBSERVATIONS

A Dissertation
Presented to
The Academic Faculty

By

Toygar Akgun

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2008

RESOLUTION ENHANCEMENT USING NATURAL IMAGE STATISTICS AND MULTIPLE ALIASED OBSERVATIONS

Approved by:

Dr. Yucel Altunbasak, Advisor
Associate Professor, School of ECE
Georgia Institute of Technology

Dr. Patricio Antonio Vela
Assistant Professor, School of ECE
Georgia Institute of Technology

Dr. Russell M. Mersereau
Regents Professor, School of ECE
Georgia Institute of Technology

Dr. Marcus Spruill
Professor Emeritus, School of Mathematics
Georgia Institute of Technology

Dr. Ghassan Al-Regib
Assistant Professor, School of ECE
Georgia Institute of Technology

Date Approved: December 5, 2007

DEDICATION

This thesis is dedicated to my dear mother, Neyhan Akgün, and to my dear departed father, Naci Sina Akgün. Thank you for your endless love and support.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Yucel Altunbasak, for his support and guidance. I also would like to thank Dr. Russell M. Mersereau, for his support and encouragement throughout my Ph.D. I am grateful to Dr. Ghassan Al-Regib, Dr. Patricio A. Vela, and Dr. Marcus C. Spruill for serving in my committee.

Many thanks go to my friends at CSIP: Ali Cafer Gurbuz, Sevgi Zubeyde Gurbuz, Soo Hyun Bae, Yeongseon Lee, Antonio Moreno-Daniel, Aytac Azgin, Ted Wada, William Mantzel and Amol Borkar. Thank you all for making CSIP such a great environment to work.

Then there are a handful that deserve to be addressed separately. I would like to thank Jenfeng (Sam) Li, for his good advice, enjoyable discussions, and most important of all, for being a great friend. I would like to thank Dr. Soner Ozgur, for teaching me a very important lesson: In life, what really counts is experience. I would like to thank Nejat Kamaci, for showing me that we do not always have to pick from what is offered to us. If we work hard enough, we can create our own options. And my deepest thanks are extended to my old group-mate, Dr. Bahadir Gunturk for introducing me to image processing, and for generously sharing his experience.

I would like to thank Dr. Khaled El-Maleh for giving me the opportunity to work in an exceptional industrial research environment at QUALCOMM. It was a real distinction and pleasure to work under his guidance. I also would like to thank Elif Albuz of NVIDIA for sharing her expertise with me.

Finally, I would like to thank my family: Naci Sina Akgün, my dear departed father, for showing me what really matters in life; Neyhan Akgün, my dear mother, for believing in me when no one else did; and Tolga Akgün, my dear brother, for showing me that true respect is not obtained by asking for it, but by earning it. In my heart, knowing that my work makes you proud, makes this thesis more than mere words and shapes on paper.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

For many digital image/video processing applications increasing the spatial resolution is highly beneficial. At higher resolution, TV pictures look more natural and pleasing to the eye, computer vision tasks such as object detection and tracking can be performed with higher precision, medical diagnoses can be made with a higher confidence, security cameras can offer better identification, and satellite imagery can be interpreted with higher accuracy. As such, spatial resolution is an influential parameter in many mainstream imaging applications, and resolution enhancement task naturally arises as a means of increasing the effectiveness of any imaging system used in the mentioned applications. In this thesis, we concentrate on two enhancement problems of practical importance, namely, low-complexity resolution enhancement for customer grade flat panel televisions, and resolution enhancement of noisy high-dimensional hyperspectral imagery. For TV resolution enhancement our main concern is keeping computational complexity at a minimum. The hardware limitations of average customer grade televisions effectively rule out a multi-frame approach. Hence, we take a low-complexity single-frame approach based on exploiting natural image characteristics. For hyperspectral imagery we take advantage of multiple observations in a modified superresolution framework. Here the main challenges are the high dimensionality of hyperspectral data and the noise present in all spectral bands. We design a physical model of the hyperspectral image acquisition process, and based on this model we formulate an iterative resolution enhancement algorithm.

Flat panel display technology is probably one of the fastest growing video display technologies with advancements taking place in all areas. Compared to the conventional cathode ray tube (CRT) displays, flat panel displays offer same screen size in much thinner forms, consume less energy, and virtually eliminate the infamous flicker problem [1] of the

[1] See Appendix A for a description of the flicker problem.

CRT displays. Perhaps the class of consumer products that enjoy these advantages the most are the customer grade TV sets. Although compactness factor and elimination of the flicker problem helped flat panel displays rapidly diffuse into the computer display market, their higher prices and lower contrast ratios (compared to CRT displays) keep them from dominating the market, at least for the time being. For customer grade TV sets, however, flat panel displays are already dominating the market. But flat panel displays have their own limitations, the most serious ones being the fixed native resolution and limited viewing angles, both of which are inherent to the pixel structure employed by flat panels. Fortunately, technologic innovations that improve the limited viewing angles are being introduced by all manufacturers. Even today, the limited viewing angle issue is slowly losing its position as the main concern of flat panel manufacturers. Fixed native resolution [2] has rather heavier implications since numerous input formats with varying spatial resolutions have to be displayed at a single fixed native resolution. Furthermore, the standard definition (SD) [3] content that dominates current TV broadcast can not be displayed on the high definition (HD) [4] flat panel displays without proper scaling. When scaled with a conventional linear shift invariant (LSI) scaling filter the resulting HD picture looks blurry with jaggy edges due to lack of high frequency components and the aliasing introduced by the linear filters with high roll-off. Hence, resolution enhancement and high quality upsampling are of utmost importance for today's HD capable flat panel TV sets.

Lately, there has been significant interest in hyperspectral imaging technology to support a variety of civilian, commercial, and military applications (civil engineering, mining, petroleum exploration, military information gathering, to name a few). The main trust behind this ever growing interest is the wealth of information in the spectral domain that provides significant advantages over traditional panchromatic and multi-spectral imagery, particularly for target and material detection, identification and classification. However,

[2] See Appendix A for a description of native resolution.
[3] See Appendix A for a description of standard definition.
[4] See Appendix A for a description of high definition.

the design and development of practical hyperspectral sensors often result in a significant tradeoff in spatial resolution. Important spatial information such as edges and texture can be lost or degraded due to reduced spatial fidelity. Unfortunately, spatial resolution is one of the most influential parameters in many hyperspectral imagery applications, and the inherent tradeoff between spatial and spectral resolution arises as a hindrance that keeps engineers from using hyperspectral data to its full extent. This observation has resulted in the development of numerous post-processing techniques to enhance spatial resolution of hyperspectral image data. There exist several techniques that can be used for spatial resolution enhancement of hyperspectral images, all of which are based on the existence of a higher resolution panchromatic or multi-spectral image obtained simultaneously by an auxiliary sensor. In this work, we present a superresolution processing based resolution enhancement technique that can increase the spatial resolution of hyperspectral data *without the requirement for high resolution panchromatic or multi-spectral image data*.

The rest of the proposal is organized as follows: Chapter 2 introduces the spatial resolution enhancement problem for customer grade TVs and hyperspectral imagery along with motivations. Chapter 3, briefly summarizes the state-of-the-art resolution enhancement techniques. Chapter 4 presents our research on single-frame resolution enhancement for customer grade TVs. Our research on hyperspectral resolution enhancement and some results are presented in Chapter 5. Finally, Chapter 6 discusses possible future research directions.

CHAPTER 2

ORIGIN AND HISTORY OF THE PROBLEM

2.1 Resolution enhancement

Let us start with defining the real world photoquantity $q(x, y, z; t; \lambda)$ as the amount of light originating from the point (x, y, z) in three dimensional space, at time t and with a wavelength of λ. Digital image/video acquisition systems can capture $q(x, y, z; t; \lambda)$ only at a fixed spatial/spectral/temporal resolution. Due to their intrinsic physical limitations, such systems are typically based on certain trade-offs between spatial, spectral and temporal resolutions. Once the image/video acquisition is performed, the obtained multi-dimensional [1] digital signal is a spatially, spectrally and temporally band-limited version of the ideal real world photoquantity $q(x, y, z; t; \lambda)$. If we consider image acquisition case, $q(x, y, z; t; \lambda)$ is projected onto the $2D$ plane of the charge coupled device (CCD) sensor, and sampled on a uniform $2D$ grid, where every sample obtained by a sensing element corresponds to a pixel value. Typically, three plane color images offer higher spatial resolution compared to multi/hyperspectral images or video at the expense of reduced spectral or temporal resolution, respectively. For multispectral and hyperspectral image acquisition, the number of spectral samples are in orders of tens and hundreds, respectively, offering a much improved spectral resolution at the expense of reduced spatial and temporal resolution. Finally, in video acquisition process a video camera samples $q(x, y, z; t; \lambda)$ on a spatio-temporal grid, where the main trade-off is between spatial and temporal resolution. In our work, we concentrate on improving the spatial resolution of image and video signals through single and multi-frame processing.

[1] Grayscale images have two dimensions corresponding to the spatial coordinates. Color images typically consist of three color planes, each of which have two spatial dimensions. Hyperspectral images have a larger number of 2D planes corresponding to different light wavelengths. Finally video signals can be interpreted as having three dimensions, where the first two correspond to the spatial dimensions, and the third dimension is time.

Spatial resolution [2] enhancement problem can be stated as estimating the high spatial frequency components of a band-limited multidimensional signal given its values on a fixed sampling grid. The resulting resolution enhanced signal needs to be rendered on a more dense spatial grid to allow proper visual representation of the estimated high frequency components. Due to this requirement, scaling (to be precise, upscaling) and resolution enhancement are interrelated, and this connection needs to be clarified. Scaling typically refers to an image processing operation that changes the dimensions of a picture or video frame [3]. Since every digital image can be interpreted as the sampled version of a continuous real-world quantity, we could also define scaling as re-sampling an input digital image on a new sampling grid, which can be more or less dense than the original grid. When the scaling ratio is smaller than one, the new sampling grid is less dense then the original grid, and the scaled image has smaller dimensions compared to the original. This case is typically referred to as zoom-out. When the scaling ratio is larger than one, the new sampling grid is more dense than the original grid, and the scaled image has larger dimensions. This case is typically referred to as zoom-in. In the scope of our work, we are more interested in scaling with ratios larger than one. As the sampling grid gets more dense and spatial dimensions are increased, the number of pixels used to render the image on a display also increase, and one may be tempted to say that the spatial resolution has been increased. At this point, it is essential to note a fundamental fact: Increasing the pixel count used to render a picture through scaling can never be equivalent to sampling the original continuous scene at the scaled (more dense) spatial sampling grid, for example, using an imaging system that has more physical sensing elements. Once a digital image is captured, the frequency content of the image is limited by the resolving power of the image acquisition system, which is a function of the density of the sensor array and the imaging optics. Scaling an

[2] The term *resolution* may have different meanings for different types of image/video data. For hyperspectral imagery both spatial and spectral resolution are meaningful, where as for TV signals spatial and temporal resolution are of interest. Through out this proposal, we use the term resolution to refer to spatial resolution.

[3] Note that for a digital image spatial dimensions are the number of samples in horizontal and vertical directions.

image by LSI filtering can not restore the high frequency components degraded (completely filtered out, reduced to noise-level or aliased) during sampling. This is where resolution enhancement differs from LSI upscaling. Single-frame resolution enhancement techniques can estimate the missing high frequency components to a limited extent through spatially adaptive filtering, and use of prior information extracted from natural images. Multi-frame resolution enhancement techniques can further improve the spatial resolution since they can recover an even larger portion of the high frequency band by fusing information embedded in multiple aliased frames. Before we conclude this discussion, we note that resolution enhancement methods can be viewed as advanced upscaling methods capable of estimating or inventing the missing high frequency signal components.

Resolution enhancement is an inherently ill-posed problem that requires extra information. In case of multi-frame resolution enhancement, typically referred to as superresolution, extra information is mainly extracted from multiple aliased observations. By precisely registering multiple observations on a common high resolution grid, multi-frame techniques extract and fuse the information embedded as aliased high frequency components. Depending on the domain in which the signal is processed, and the filtering technique used to compute the final high resolution frame, different superresolution methods are obtained. In case of single-frame resolution enhancement we do not have access to multiple frames, hence we are bound to use prior information. Prior information can be in the form of *a priori* distributions in the Bayesian framework or regularization terms in the deterministic approach. Another way of utilizing prior information is to learn a group of spatial structures (which we refer to as context classes) frequently observed in natural images and observe the way they are distorted during high resolution to low resolution conversion (sampling or downsampling). There are at least two well-known single-frame resolution enhancement algorithms that utilize prior information in this format, namely, resolution synthesis proposed by Atkins *et. al.* [1] and example-based superresolution by Freeman *et. al.* [2]. Resolution synthesis (RS) is based on pixel classification and adaptive

linear filtering, and allows for efficient hardware implementation. Since, our goal is to design a low-complexity resolution enhancement method that can be implemented in the next generation display systems, we focus on the RS algorithm.

2.2 Low-complexity single-frame resolution enhancement

With the introduction of high definition television (HDTV) the visual quality of the customer grade TV sets have increased substantially. HDTV offers almost a two times increase in spatial resolution compared to the the NTSC and PAL/SECAM standards of the standard definition television (SDTV). While a number of HDTV standards have been proposed, the current HDTV standards as defined in ITU-R BT.709 have screen resolutions of 1080×1920 (progressive or interlaced scan [4]) or 720×1280 (progressive scan [5]). The screen sizes of HD displays supporting these resolutions are much larger than the TV sets used to view SDTV. As we discussed in Chapter 1, thinner form factor and elimination of the flicker problem make flat panel displays more preferable over the CRT displays for larger screen sizes (40 inhes or more). Due to their fixed native resolutions, flat panel displays require high quality scalers to scale a variety of input resolutions to the screen's native resolution. Furthermore, most of the existing content and the current TV broadcasting is in SD, and needs to be properly scaled to be displayed on HD displays. These facts make scaler arguably the most important block in the video pipeline of any flat panel TV, with a huge influence on the success of the product.

Unfortunately, TV sets typically have very strict constraints on the on-board hardware, hence the complexity of scaling algorithms that can be implemented in TV sets is severely restricted. The amount and complexity of the hardware components in the video pipeline has a direct influence on unit price, and unit price is the most influential parameter for comparable TV sets fitting in the same product profile defined by factors such as the screen size, picture quality, form factor and target customer group. Naturally, all manufacturers

[4] For a definition of the interlaced scan see Appendix A

[5] For a definition of the progressive scan see Appendix A

focus on producing TV sets at the lowest cost, while still achieving certain picture quality standards. These picture quality standards are carefully extracted from experimental studies conducted on large groups of customers that represent the average viewer. These studies are based on the observation that many customers do not even possess the educated eye or know-how to accurately differentiate between products of comparable picture quality. In short, what really matters is not producing the TV with the best picture quality. The real competition is to design a TV that matches certain picture quality standards at the lowest unit price. The bottom line on our side is a strict limitation on the computational complexity of our resolution enhancement algorithm. Hence, we eliminate multi-frame techniques and focus on single-frame resolution enhancement.

LSI scalers such as nearest neighborhood, bilinear and bicubic filters offer mediocre visual quality at low computational complexity. Although these scaling techniques are the industry standard, especially for low end products, their visual quality is typically plagued by two problems, namely, blur and jagged edges. Blur is a clear indication that we are not fully utilizing the spatial resolution offered by the display. In case of SD to HD conversion, this is mainly caused by the fact that SD definition content is optimized for lower resolution SD displays. To avoid aliasing, SD content is typically anti-alias filtered during capture and/or during post-processing, resulting in degradation of high frequency signal components. Jagged edges, on the other hand, are mainly caused by the large roll-off factors of linear interpolation filters. Since these filters are too short to provide high enough rate of decay during transition from the pass-band to the stop-band, they leak the high frequency components of the neighboring frequency domain replicas into $[-\pi, \pi]$ creating superficial high frequency "details".

The improved native resolution and large screen sizes offered by HD displays are both a blessing and a curse. The increased spatial resolution can render much more pleasing pictures. But at the same time, artifacts such as blurred or jaggy edges are made more visible, and can be detected even by uneducated eyes of an average viewer. Hence, we

require our scaler to produce high quality, artifact-free pictures for a variety of inputs such as natural scenes, text characters, and interlaced video signals. Special care must be taken to handle these cases properly, and assure that the scaler does not create any artifacts easily detectable by an average viewer. As detailed in Chapter 4, we design our scaler within an improved RS framework obtained by modifying the existing framework to better suit our needs.

2.3 Hyperspectral imaging and spatial resolution enhancement

Any physical object in a scene reflects, absorbs and emits electromagnetic radiation. The object's molecular composition and shape affect the way this interaction occurs. Using this phenomenon to gather information about an object or scene without coming into physical contact with it is called *electro-optical remote sensing*. If the electromagnetic radiation arriving at the sensor array is measured at a sufficiently high number of wavelengths for every pixel, the resulting spectrum can be used to extract information that cannot be extracted from images captured by conventional devices that do not provide much information about the spectral dimension. Topics involved with the measurement, analysis and interpretation of such spectra are treated in the field of *spectroscopy*. Another related field, *imaging spectroscopy*, combines spectroscopy with methods to acquire spectral information.

Microwave, RADAR, thermal infrared, ultraviolet and multi-spectral sensing instrumentation have been successfully used for remote sensing applications. But the most significant recent breakthrough in the field of remote sensing has been the development of hyperspectral sensors. Hyperspectral sensors are a class of imaging spectroscopy sensors, for which the sensed waveband is divided into *hundreds* of contiguous narrow frequency bands. As the name suggests, hyperspectral sensors differ from their predecessors, the multispectral sensors, in that the number of bands that are separately imaged is much higher. (For example, the AVIRIS, Airborne Visible/Infrared Imaging Spectrometer, from NASA/JPL has 224 bands.) Hyperspectral sensors commonly produce images in 12 to 288

separate bands, usually covering the region from 400 to 2500 nanometers. Over the past two decades hyperspectral image analysis has matured into one of the most powerful and fastest growing technologies in remote sensing. In 1983, NASA flew an experimental sensor system called the Airborne Imaging Spectrometer (AIS) with 128 bands. Then, in 1987, the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) with 224 usable bands was flown. There are also several other commercial systems such as the MIVIS (Multispectral IR and Visible Imaging Spectrometer) and AHS (Airborne Hyperspectral Scanner) systems manufactured by Daedalus Enterprize Inc. For a detailed treatment of hyperspectral imaging the interested reader can refer to [3] and [4].

The wealth of spectral information provided by hyperspectral sensors implies significant advantages over traditional panchromatic and multi-spectral imagery, particularly for ground target and material detection, identification and classification. *However, the increased spectral resolution provided by hyperspectral sensors comes at the expense of reduced spatial resolution.* The design and development of practical hyperspectral sensors often result in a significant tradeoff in spatial resolution. A high-resolution hyperspectral sensor is difficult to develop and deploy because of the requirement for a sufficiently large and accurate optical system to collect the light. Consider, the Hyperion hyperspectral sensor on the NASA EO-1 spacecraft, which offers the full reflection band from 450 to 2300 nm at 30-meter spatial resolution. [6] Hyperspectral sensors require greater sensitivity than typical sensors in order to overcome the reduced photon count in narrow wavelength bands. Higher sensitivity is achieved by larger apertures ($\sim 0.3m$), large pixel cell ($\sim 100mm$), and slower operating speeds (between 1 and 10Hz), as well as low detector temperature ($\sim 10K$) to suppress the effects of dark current. The Hyperion sensor has a 12.5 cm aperture and weighs approximately 50 kg. To simplify operations and keep the package size weight and power to a minimum, Hyperion acquires data by push-brooming along its ground track

[6] The resolution of many satellite-based instruments ranges from about 10 meters to several kilometers. Defense/military satellites typically have higher spatial resolutions but exact specifications are classified and not available for widespread use. Commercial satellites planned for launch in the near future will carry sensors with spatial resolutions of 5 meters or less.

at the orbital rate without any image motion compensation. If we scale this to a system that could acquire data at four times the spatial resolution (at 7.5 m spatial resolution), the aperture would have to be increased by a factor of four in diameter (to 50 cm) with a comparable increase in focal length and complexity of the optics. If we assume that the weight scales as the square of the aperture, the sensor weight would increase to about 800 kg.

The inherent tradeoff between spatial and spectral resolution has resulted in the development of post-processing techniques to enhance spatial resolution of hyperspectral image data. There exist several approaches that can be used for hyperspectral resolution enhancement, all of which are based on the existence of a higher resolution panchromatic or multi-spectral image obtained simultaneously by an auxiliary sensor [5], [6], [7]. In this work, we use a modified superresolution (SR) reconstruction technique to enhance the spatial resolution of AVIRIS data *without the requirement of high resolution panchromatic or multi-spectral auxiliary images*. SR reconstruction is a multi-frame signal processing technique capable of fusing information that is embedded in multiple aliased observations of a scene. Applications of SR include aerial and satellite imaging, hyperspectral imaging, security and surveillance, forensic science, up-conversion of digital TV (DTV) and NTSC signals, printing from video, and digital cameras among others. Perhaps, the most well-known SR application is the generation of high-resolution images through the fusion of several low-resolution images captured by the *Mars Pathfinder* cameras. Similarly, in the area of aerial photography, several researchers have shown that low-resolution frames captured through an array sensor on-board an aircraft can be processed to construct higher-resolution aerial maps [8]. In intelligence-applications, despite the high-resolution custom design cameras/sensors on board spy satellites and aircraft, some objects may still not be properly resolved due to the vast distances involved and atmospheric degradations, and application of SR algorithms is required for deeper analysis. Security and surveillance systems usually employ low-resolution sensors to minimize the cost and storage requirements. Hence, subsequent SR processing to resolve the important details may be needed in the

case of an event. Likewise, forensic science employs SR methods to solve criminal cases by identifying the texts, marks, etc. hidden within low-resolution video.

Application of SR techniques to hyperspectral data presents several research challenges, the most important ones being the increased computational complexity and noise amplification as we discuss in Chapter 5. Since the resulting resolution enhanced images are typically input to a target detection, identification or tracking engine, we were attracted to investigate the possibility of pre-processing hyperspectral in such a way that only the portion that is of interest is enhanced by the SR restoration block. This approach resulted in a modified superresolution technique that operates on transformed data. Our tests confirms that our modified SR technique has two main advantages over processing full data. First, by projecting the data over a subspace that is optimized for certain spectral signature of interest, we can effectively minimize the undesired effects of signal noise. Second advantage is the reduction in computational complexity. A detailed derivation of the proposed resolution enhancement technique together with visual results and in depth discussions will be presented in Chapter 5.

CHAPTER 3

RELATION TO THE STATE-OF-THE-ART

3.1 Single frame resolution enhancement: resolution synthesis

Natural images are highly structured signals with much less variability than completely random signals of the same dimensionality [9], [10]. As a result of this structure and highly constrained spatial variation, natural images exhibit certain spatial characteristics that can be exploited for resolution enhancement purposes. Based on this observation, the main idea behind single-frame resolution enhancement is to exploit prior information extracted from large sets of training images.

In general, single-frame resolution enhancement methods consist of two processing stages, namely, training and filtering. Typically, training stage is computationally intense, and is performed off-line. In the training stage, a large set of hand-picked images are processed to extract prior information about the spatial image characteristics. Prior information [1] consists of a group of learned spatial structures (which are typically referred to as *context classes*) frequently observed in natural images, and the ways they are distorted during high resolution to low resolution conversion. By modeling the learned spatial structures, we can design ways to recognize them in input images that were not even in the original training set. By investigating the class specific distortions, we can design filters to restore the degraded image components. In the filtering stage, the input pixels are first classified into one of the learned context classes depending on their spatial structure, and the corresponding reconstruction filter is applied. There exist at least two single-frame resolution enhancement methods that utilize prior information in this form, namely, *resolution synthesis* proposed by Atkins *et al* in [1], and *example-based superresolution* proposed by Freeman *et al* in [2].

[1] It is also possible to extract prior information in the form of an *a priori* distribution on the pixels or pixel structures observed in natural images. The learned prior can be used in a Bayesian framework.

Resolution synthesis is based on pixel classification and adaptive linear filtering. Context classes are modeled as a multivariate Gaussian mixture, whose parameters are estimated from the training data. In the training phase we look at many high-low resolution pixel pairs and extract frequently observed spatial structures defined in small local neighborhoods. Examples of such local structures are edges of various orientations, texture and uniform areas. Since we have access to both high and low resolution versions of the pixel groups, it is possible to learn how the high resolution neighborhoods are degraded during the downsampling process. After grouping many low resolution pixels classified as belonging into a certain context class, computational optimization techniques are used to obtain the corresponding optimal inverse filter (*i.e.*, the optimal interpolation filter) by minimizing a cost function designed to access the similarity of the reconstructed high resolution pixels to the original high resolution pixel values. When a new low resolution pixel is given, the algorithm first looks at a local pixel neighborhood to decide on which filter to use (*i.e.*, *classification*). Then interpolation is performed by applying the optimal linear filter (*i.e.*, *filtering*).

Example-based superresolution proposed by Freeman in [2] is different from resolution synthesis in its pixel classification, context modeling and reconstruction methods. In the example-based superresolution framework, the high resolution image is synthesized block by block, where the blocks are chosen from a database constructed during the off-line training phase. Example-based superresolution first decomposes the input image into low frequency (*mean*) and high frequency (*detail*) components. The mean image is simply LSI scaled. The detail component of the output is obtained by stitching restored high resolution patches together, where stitching is performed by classifying the low resolution details. The restored high resolution patches are learned from a training set together with a statistical model that explains the local pixel relationships used to decide which high resolution patch goes where. The resulting scaled *mean* and high resolution *detail* images are combined to obtain the final high resolution output.

When these methods are compared in terms of computational complexity and ease of hardware implementation, resolution synthesis is found to offer numerous advantages over example-based superresolution. Example-based superresolution requires a data transformation as an initial processing stage, introducing additional computations. Furthermore, to perform reconstruction we have to store all the high resolution patches learned during training, which requires large amounts of memory. Since we aim for a computationally efficient resolution enhancement method that can be implemented in low to mid-end customer grade TV sets, we focus on resolution synthesis.

3.2 Multi-frame resolution enhancement: superresolution

The main idea behind superresolution is fusing information that exists in a sequence of noisy, blurred and aliased images to produce an image or a sequence of images of higher spatial resolution. The information required to reconstruct the missing or degraded high frequency components is embedded in the low resolution input images/frames as aliasing. In their early work on the subject, Tsai and Huang [11] disregarded the blur in the imaging process and carried out a frequency domain analysis of the superresolution problem. They showed that any effective superresolution method requires frequency aliasing to be present in the low resolution observation (source) images. Below we present a review on the state-of-the art in superresolution research. A comprehensive background on superresolution can be found in [11] and [12].

Superresolution reconstruction methods consist of three main stages, namely, registration, non-uniform interpolation and restoration. Image registration is the process of mapping low resolution pixels from all available observations to a common high resolution reference frame. Non-uniform interpolation step converts the non-uniform sampling lattice obtained by the image registration block to a uniformly sampled high resolution grid. Finally, the restoration block removes blur and noise introduced during image acquisition

process. These processing blocks can be implemented simultaneously or separately. Existing superresolution methods can be classified depending on the mathematical tools they employ, the domain they operate on and the type of image data they are designed for.

Non-uniform interpolation is the most straight-forward approach to superresolution. Image registration, non-uniform interpolation and restoration blocks are implemented separately, and performed successively to obtain the high resolution reconstructed image. The main advantage of the non-uniform interpolation methods is their low computational complexity and ease of implementation. However, these methods are only applicable if the blur and noise characteristics are the same for all observations. Examples of this approach are Ur and Gross [13], Komatsu *et al.* [14], Hardie *et al.* [15], Shah and Zakhor [16].

Frequency domain approach makes explicit use of the aliasing that exist in the low resolution observations. The most important short coming of frequency domain methods is their inherent limitation to global translational motion. The original frequency domain approach, which did not include blur and noise present in the low resolution observations, was detailed by Tsai and Huang in [17]. Kim *et al.* [18], [19] extended [17] to a least squares problem, where noise and blur present in the low resolution observations were explicitly taken into account. Tom *et al.* [20], [21] proposed to solve image registration and restoration problems simultaneously through the use of expectation-maximization algorithm in the frequency domain.

Superresolution reconstruction is an ill-posed problem because of the insufficient number of observations and ill-conditioned blur operators. Stabilization of ill-posed inverse problems through redefinition of the solution to impose certain desired properties known to exist in ideal high resolution images is referred to as regularization. Typically, regularization framework is based on the minimization of a cost function designed to penalize solutions that do not agree with the observations *and* do not have desired properties such as smoothness. Examples of this approach are Katsaggelos *et al.* [22], Hardie *et al.* [23], and Bose *et al.* [24].

The iterative convex projection based methods provide a flexible and intuitive way of incorporating prior knowledge about the solution into the reconstruction process. Every registered low resolution pixel (together with a forward imaging model) defines a constraint set. Additional constraint sets can be defined based on the prior information about the desired solution, such as positive pixel values and the allowed range of $[0, 255]$. To obtain the reconstructed image an initial estimate is iteratively projected on the convex constraint sets until some convergence criterion is met. The projection onto convex sets (POCS) formulation of superresolution was first suggested by Stark and Oskoui [25]. Their work was extended by Tekalp [26], [27] to include observation noise by using a spatial noise variance estimate on a pixel-by-pixel basis to directly constrain the solution. In [28] Patti *et al.* proposed a POCS based superresolution method where a continuous image formation model is developed to allow for higher order interpolation methods.

Statistical estimation based superresolution approach computes the high resolution image as the maximum likelihood (ML) or maximum *a posteriori* (MAP) estimate under some statistical model. In [29] Schultz and Stevenson, and in [30] Stevenson and Schmitz described a MAP estimator with a Huber-MRF (Markov random field) prior model to preserve discontinuities and solve the blurring problem introduced by imposing smoothness. Depending on the statistical model of the observation noise, and the *a priori* image model, the resulting methods can be computationally intense. It is also possible to combine POCS based methods with ML or MAP estimation. The resulting ML/MAP-POCS hybrid approach computes the superresolution estimate by minimizing the ML/MAP cost functional while constraining the solution within certain convex sets imposed by the desired properties of the solution.

In general, superresolution algorithms try to regularize the ill-posedness of the problem by fusing information from multiple frames and using prior knowledge about the solution, such as smoothness or positivity [31]. Recently, researchers have proposed algorithms that attempt to use model-based constraints in regularization. If the images to be super-resolved

consist of a restricted set with well-defined spatial structure (such as face images) then the characteristics inherent to that group of images can be exploited. In this context, superresolution techniques have been proposed for face recognition that attempt to obtain a high-resolution face image by combining the information from multiple low-resolution images, [32], [33], [34], [35], [36]. While [32] demonstrates how superresolution (without model-based priors) can improve the face recognition rate, [33], [34], [35], and [36] provide superresolution algorithms that use face-specific constraints for regularization. Gotoh and Okutomi [37] proposed a superresolution method aimed for images obtained by a single-CCD with a color filter array. Their method is based on a generalized formulation of superresolution which performs both resolution enhancement and demosaicking simultaneously, and is capable of producing a high-resolution color image directly from color mosaic images obtained by a single-CCD with a color filter array. Superresolution reconstruction techniques have also been successfully applied to multi-spectral [38] and hyperspectral imagery [39] to improve the accuracy of object detection, identification and tracking algorithms.

In the area of compressed-domain processing, Chen and Schultz [40] applied spatial-domain SR methods to *decoded* MPEG frames by utilizing MPEG motion vectors to initialize the motion estimation phase. Recently, Mateos, Katsaggelos and Molina [41] proposed a MPEG-compressed video enhancement algorithm. In [42] Patti *et al.* develop a superresolution reconstruction method that can operate on compressed video sequences directly.

Wavelet domain superresolution methods are based on the multi-resolution analysis framework provided by the wavelet transform that can decompose signals into components at different scales or resolutions. In [43] Nguyen and Milanfar presented a superresolution method that exploits the interlacing structure of the sampling grid. Using a separable orthonormal wavelet basis for 2D images, they derived a wavelet decomposition using Kronecker products, resulting in an efficient calculation of the wavelet coefficients. In [44], [45]

Chappalli *et. al.* pointed out that second generation wavelets are better suited for superresolution, and proposed a method based on second generation wavelets that perform simultaneous denoising and superresolution. Wavelets and multi-resolution analysis are especially well-suited for astronomical image processing because they are adept at providing accurate, sparse representations of images consisting of smooth regions with isolated abrupt changes or singularities (e.g. stars in dark sky). Based on this observation Nowak *el al.* [46] developed a wavelet-based superresolution method for astronomical imagery. Their proposed approach uses the expectation-maximization algorithm for superresolution image reconstruction based on a penalized likelihood formulated in the wavelet domain.

Note that the boundaries separating these approaches from each other may not be clear for all cases. For example, statistical estimation based methods such as maximum *a posteriori* estimation can be interpreted as an alternative way of regularization, and it is possible to formulate regularization or statistical estimation approaches in spatial or frequency domain.

3.3 Contributions

Our contributions can be collected under two main titles, namely, single-frame resolution enhancement (Chapter 4) and multi-frame resolution enhancement as detailed (Chapter 5). In this section we summarize our contributions.

In the area of single frame resolution enhancement our main contribution is algorithmic and computational improvement of the resolution synthesis method. Our main goal was to investigate the use of resolution synthesis as a high quality low computational complexity resolution enhancement method for customer grade TV sets. Our research on the original algorithm pointed out several shortcomings in the feature extraction and off-line training stages. We designed and tested separate improvements for each case. Based on the observation that spatial aliasing and noise have a devastating effect on feature extraction and pixel classification stages, we designed a better feature extraction block that provided

improved visual performance. To improve the off-line training we introduced a coupling between the feature extraction and filter design stages with the ultimate goal of propagating high resolution spatial information to feature extraction stage. Our efforts resulted in an iterative off-line training algorithm that provided improved visual performance. Finally, a major shortcoming of the original resolution synthesis algorithm was its limitation to integer scaling ratios. We extended the resolution synthesis method to handle non-integer scaling ratios. Our improved algorithm was implemented in Xilinx Spartan 3 FPGA board, and the implementation details were published in [47].

In the area of multi-frame resolution enhancement our main contribution is the adaptation of superresolution technique to hyperspectral imagery. Most of the existing resolution enhancement techniques for hyperspectral imagery are based on the existence of a high resolution panchromatic or multi-spectral observation. Motivated by the fact that hyperspectral observations typically obey affine and translational motion models, we proposed to apply superresolution to hyperspectral imagery. We formulated a complete hyperspectral imaging model capable of incorporating any linear spectral representation model and spatial blurring effects. Since principal component analysis (PCA) is a popular processing step in several hyperspectral imaging applications, we first incorporated a PCA based spectral representation model. Based on the resulting imaging model we formulated and implemented spatial superresolution for hyperspectral imagery. The most obvious shortcoming of the PCA based spectral representation is its incapability to represent specific spectral signatures of interest. Since PCA is designed to capture most of the variation contributed by all spectral bands at any spatial location, it can not emphasize single spectral signatures. To remedy this shortcoming we next modified our imaging model to incorporate the linear spectral mixing model. Based on the updated model we formulated and implemented material specific multi-frame resolution enhancement for hyperspectral imagery.

CHAPTER 4

SINGLE FRAME RESOLUTION ENHANCEMENT

As we discussed in Section 2.2 our goal is to design a low-complexity single-frame resolution enhancement algorithm that allows for an efficient hardware implementation. Since we are aiming for customer grade TV sets and set-top boxes, keeping computational complexity at a minimum is of utmost importance. We can tolerate heavy off-line computations, but filtering needs to be performed in real-time, and must be as computationally simple as possible. In the light of our discussions in Section 2.2 we base our approach on the resolution synthesis framework. We start with analyzing resolution synthesis as proposed by Atkins in [1]. Then we point out several short-comings that avoid an efficient hardware implementation. Finally, we present our solutions to these shortcomings, and provide visual comparisons of the improved and original resolution algorithms.

4.1 Resolution synthesis framework

The main idea behind resolution synthesis (RS) can be stated as: *In a large training set, learn the high resolution image details that correspond to different spatial structures observed at low-resolution, such as edges of different orientations, uniform areas and texture regions, then use those learned relationships to identify and restore the details in other images.* To get a better grip on the idea, note that natural images are highly structured signals with much less variability than completely random images. The regularities typically observed in natural images [9] can be exploited in the resolution enhancement problem. However, we point out the fact that unless we are working on a highly restricted set of images with very specific training data, it is not possible to generate the *true* high-resolution signal components. Hence, instead of trying to reconstruct the true high resolution details, we focus on generating *visually plausible* image details, such as sharp edges without disturbing jaggies, and natural looking texture. In other words, rather than maximizing fidelity

to the true signal degraded during acquisition, we aim to maximize subjective visual quality by restoring the details most likely to be found in natural images.

Resolution synthesis approach is based on recognizing that pixels in natural images can be classified as belonging into a limited number of context classes. These context classes are defined by small local pixel neighborhoods that exhibit visually identifiable spatial structures. Figure 4.1 shows three 5×5 local neighborhoods that can be identified as belonging into a vertical edge class.

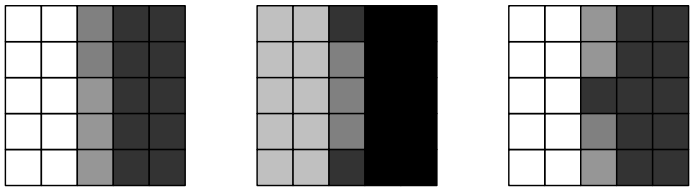


Figure 4.1. Several 5×5 pixel neighborhoods that can be identified as vertical edges

To identify such context classes we prepare a large training set consisting of properly registered high-low resolution image pairs. Depending on the size of the local window, the number of all possible pixel patterns can be very large. For example, considering a 5×5 window and 8-bit gray-scale images, the number of possible pixel patterns is

$$\underbrace{2^8 \times 2^8 \times \ldots 2^8}_{25} = 2^{8 \times 25} = 2^{200} \approx 10^{60}. \tag{4.1}$$

We can not possibly handle such a large number of classes, effectively eliminating the option of assigning a context class to every possible pixel configuration. Fortunately, for natural images, the frequency of a large proportion of these possible patterns are very low, simply because they do not really correspond to meaningful spatial structures. In other words, a great proportion of these possible patterns are highly random. As such, they exhibit a noise-like behavior, and can not appear frequently in natural images.

Even if we had a means of eliminating such noise like patterns, we would still be stuck with a large number of possible pixel configurations. At this point, let us have another look at Figure 4.1. Although these neighborhoods are not identical pixel by pixel, they all

have the same dominant spatial structure, which is a vertical edge. Our intuition tells us that if we were to derive the optimum interpolation filters for these pixel neighborhoods, the resulting filters would be very close [1]. In the light of this observation, we see that instead of assigning a context class to every possible pixel configuration, we can have a small number of context classes each of which represents a large number of pixel neighborhoods with similar spatial structure. This can be achieved by clustering similar pixel neighborhoods into a single representative context class. Typically, the number of context classes required to obtain the best clustering is unknown. If the number of classes is fixed, we can always find the best fitting partition (either crisp or fuzzy) of the data by applying a clustering method such as c-means, fuzzy c-means, Gustaffson-Kessel, or expectation-maximization based on Gaussian mixture assumption. But we can always achieve a better fit by increasing the number of classes, and the real challenge becomes finding the minimum number of classes over which increasing the class number does not substantially improve the fit to the data[2]. There exist metrics designed to measure the goodness of fit to a given data set [49], [50] (hence, the quality of the obtained clustering result), and clustering schemes that adaptively change the number of clusters based on these metrics to remedy the unknown cluster number problem. In the original resolution synthesis framework described in this section, the suggested number of classes is around 100. However, in our case the maximum allowable number of classes is dictated by hardware constraints. After an extensive study of the available resources [3], we have decided to limit the number of context classes to eleven. Keeping the visual performance at a satisfactory level while reducing the number classes to eleven requires serious improvements in clustering, feature extraction, and classification blocks. Our proposed improvements will be discussed in the next section.

For clustering purposes, every low resolution pixel is represented by a feature vector $\boldsymbol{y}$

[1] Given some optimality criteria, there exists tools and algorithms to compute the optimum filters. For example, in [48] Li *et. al.* describe a method to compute the interpolation filters optimal in the MSE sense.

[2] Note that using a large number of classes can lead to overfitting.

[3] We would like to offer our thanks to VESTEL Corporation for providing information about the hardware limitations of the current TV sets.

extracted from its local neighborhood [51]. By carefully designing the feature extraction function, we can enhance the clustering performance compared to using the pixel values directly. Note that the feature extraction function can be designed to nonlinearly distort the space of the pixel values to accentuate certain spatial features of interest. For example, in scaling rendition of edges is of great importance. By carefully selecting the feature extraction function, we can emphasize edges so that they are better separated from other image features such as texture and uniform regions. To accurately differentiate between different context classes feature vectors must be good representatives of the neighborhoods they are extracted from. If this is not the case, both the training and the classification phases suffer from faulty classifications. For the training phase, misclassified neighborhoods distort the extracted prototypes degrading the classification performance. For the filter design stage, the optimization method will erroneously try to fit the filter to the misclassified neighborhoods, degrading the performance for the true class.

Figure 4.2 shows a 5×5 local window centered at the current input pixel. Using the given pixel naming convention, we first obtain a 8×1 vector $\tilde{\boldsymbol{y}}$ as in Figure 4.2. Then

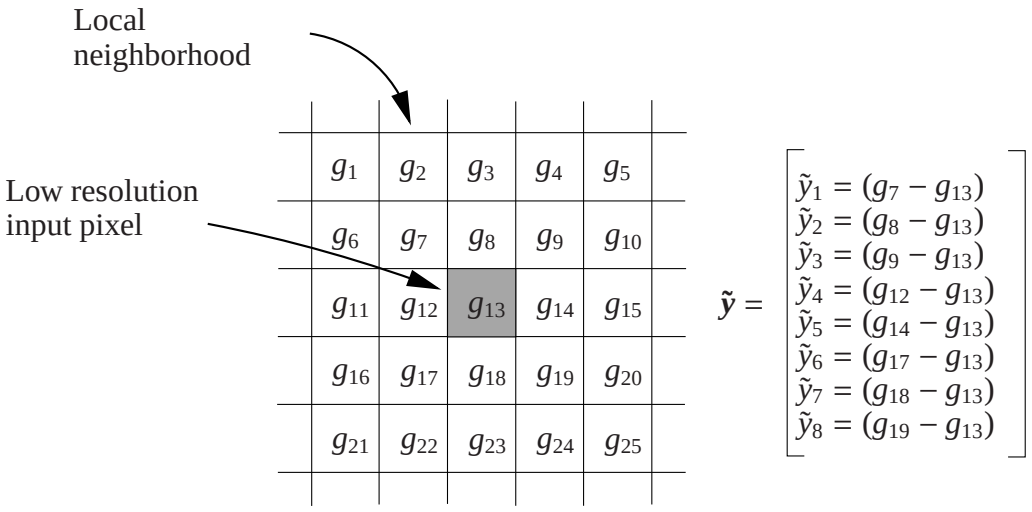


Figure 4.2. 5×5 **low resolution feature extraction neighborhood**

the feature vector $\boldsymbol{y}$ proposed by Atkins in [1] is computed as the normalized version of $\tilde{\boldsymbol{y}}$,

where the elements of $\boldsymbol{y}$ are given as

$$y_i = \frac{\tilde{y}_i}{\left[\sum_{j=1}^{8} \tilde{y}_j^2\right]^p} \quad . \tag{4.2}$$

Here $0 \leq p \leq 1$ is a parameter that controls the amount of normalization. For $p = 1$ all the feature vectors are normalized to unit length. For smaller values of p normalization is less severe. Normalizing all feature vector to unit vector has the undesired effect of compressing all vectors onto the 8-dimensional hypersphere. Under this normalization, feature vectors are so densely packed that separating feature vectors belonging to different context classes becomes harder. At the other extreme, if we do not have any normalization, even the feature vectors coming from similar local neighborhoods may have high variations in their lengths. Hence, p is typically chosen as 0.75, a value that has been shown to provide a good trade-off [51].

Once we have the feature vectors, we can cluster them to come up with a relatively small number of representative context classes. This is achieved by modeling the feature vectors as a random vectors drawn from a multivariate Gaussian mixture with M mixture classes, where every Gaussian mixture class corresponds to a context class. The expectation-maximization (EM) algorithm is applied to compute the maximum likelihood (ML) estimates of the Gaussian mixture parameters, namely, the class means ($\boldsymbol{\mu}_i$), standard deviation (σ) [4] and mixture probabilities (π_i). Once EM converges and the maximum likelihood estimates of the Gaussian mixture model are obtained, we can compute the probability that any given feature vector belongs to a mixture (context) class. If these probabilities are interpreted as memberships than the resulting mixture model provides a fuzzy clustering of the feature vectors in the training set. New input pixels are classified by computing the probabilities that their feature vectors are drawn from a context class.

Let us denote the raster-scanned high resolution pixels by $\boldsymbol{f}$, and the raster-scanned low resolution pixels by $\boldsymbol{g}$. Derivation of the optimal resolution synthesis filters is based on the

[4] The same σ is used for all classes and for all feature vector entries. This is a quite limiting assumption and will be discussed in Section 4.2.

following assumptions:

- **Assumption 1**: Feature vectors are modeled as a multivariate Gaussian mixture

$$p_Y(\boldsymbol{y}) = \sum_{j=1}^{M} p_{Y|J}(\boldsymbol{y}|j)\pi_j, \qquad p_{Y|J}(\boldsymbol{y}|j) \sim \mathcal{N}(\boldsymbol{\mu}_j, \sigma^2\boldsymbol{I}).$$

- **Assumption 2**: Given the input low resolution pixel neighborhood and the context class, the high resolution pixels are Gaussian

$$p_{F|G,J}(\boldsymbol{f}|\boldsymbol{g}, j) = \mathcal{N}(\boldsymbol{A}_j\boldsymbol{g} + \boldsymbol{\beta}_j, \sigma^2\boldsymbol{A}_j^T\boldsymbol{A}_j).$$

- **Assumption 3**: Given feature vector $\boldsymbol{y}$, the class distribution is independent of the high resolution and low resolution pixels

$$p_{J|F,G}(j|\boldsymbol{f}, \boldsymbol{g}) = p_{J|Y}(j|\boldsymbol{y}).$$

Under these assumptions the MMSE estimator can shown to be [51]

$$\begin{aligned}
\hat{\boldsymbol{f}} &= \mathcal{E}[\boldsymbol{f}|\boldsymbol{g}] \\
&= \sum_{j=1}^{M} \mathcal{E}[\boldsymbol{f}|\boldsymbol{g}, j]p_{J|G}(j|\boldsymbol{g}) \\
&= \sum_{j=1}^{M} (\boldsymbol{A}_j\boldsymbol{g} + \boldsymbol{\beta}_j)p_{J|Y}(j|\boldsymbol{y}) \qquad (4.3) \\
&= \sum_{j=1}^{M} (\boldsymbol{A}_j\boldsymbol{g} + \boldsymbol{\beta}_j) \underbrace{\frac{\pi_j \exp(-\frac{1}{2\sigma^2} \parallel \boldsymbol{y} - \boldsymbol{\mu}_j \parallel^2)}{\sum_{i=1}^{M} \pi_i \exp(-\frac{1}{2\sigma^2} \parallel \boldsymbol{y} - \boldsymbol{\mu}_j \parallel^2)}}_{w_j}, \qquad (4.4)
\end{aligned}$$

where $\mathcal{E}[\cdot]$ denotes the statistical expectation operator, $\boldsymbol{A}_j$ is the optimal filter for class j, and $\boldsymbol{\beta}_j$ is the optimal bias term for class j. From Eq. 4.3 we can see that final high resolution pixel estimates are computed as a weighted linear combination of the estimates for all context classes. Hence, resolution synthesis as shown in Figure 4.3 consists of three processing stages.

- **Step 1**: Perform off-line training (performed only once, before the classification and filtering steps). Apply EM to get the ML estimates of model parameters, $\boldsymbol{\mu}_j$, π_j, and σ for $j = 1, \ldots, M$.

- **Step 2**: For these estimates, compute the ML estimates of the filter parameters, $\boldsymbol{A}_j$ and $\boldsymbol{\beta}_j$ for $j = 1, \ldots, M$.

- **Step 3**: For every input low resolution pixel, extract the feature vector, compute the weights (w_j), and use the optimized filters to get the high resolution estimate as the weighted linear combination of the estimates for all context classes.

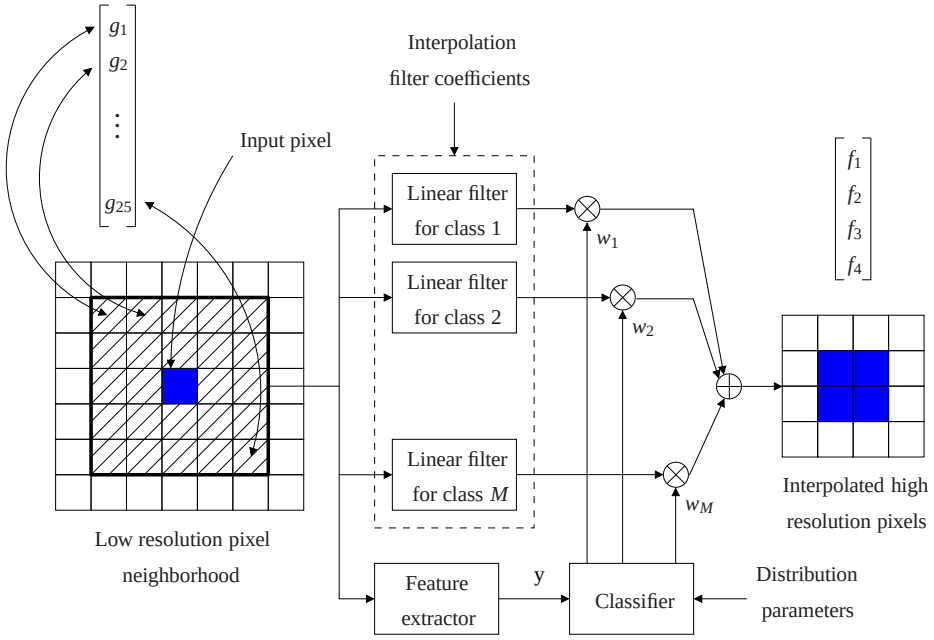


Figure 4.3. Resolution synthesis block diagram

We conclude this section with an example that demonstrates the potential visual improvement offered by context dependent filtering. In this example we upsample the image shown in Figure 4.5(a) by four, using three different interpolation filters, namely, the bilinear interpolation filter, a 91×91 separable linear interpolation filter, and a 12×12 nonseparable (directional) linear interpolation filter [5]. The 91×91 separable filter was designed

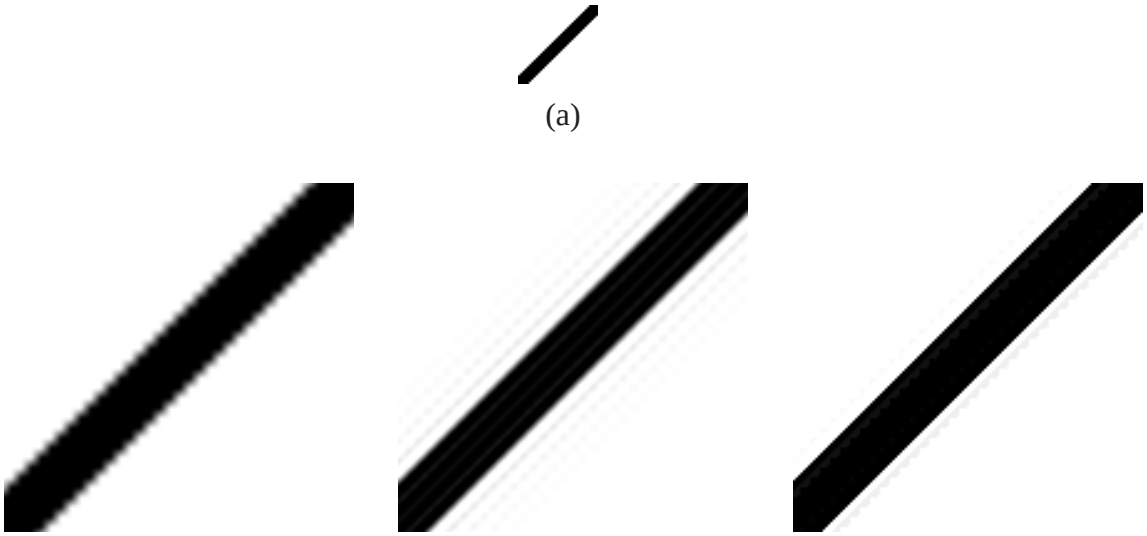
[5]The 12×12 mask includes all phases of the directional filter. Since we have 4 phases for each direction,

to have a quite precise cut-off frequency of $\pi/4$ with negligible ripple in both pass and stop-bands. The separated 1D frequency response, which is identical for both vertical and horizontal directions, is shown in Figure 4.4(a). The 2D frequency response obtained by convolving the 1D filter in vertical and horizontal directions is shown in Figure 4.4(b). The directional filter was obtained by resolution synthesis configured to have a single context class. Note that the input image was carefully chosen to accommodate this scenario. The training was done on the picture shown in Figure 4.4(a) and its four times downsampled version. The 2D frequency response of the directional filter is shown in Figure 4.6. Figures 4.5 (b),(c) and (d) show the upscaling results. Bilinear interpolation output is blurry with jaggy edges. The 91×91 separable filter improves the jaggies but still has a blurry look with visible ringing. Finally, the 12×12 nonseparable directional filter provides sharp transitions with reduced ringing using a much smaller filter size. Of course, in this example we have a single context class, and faulty classification is not an issue. This points out an important aspect of training based scaling methods: *The most essential part of the algorithm is the classification.* As long as classification performs fairly good, training and filtering stages can be optimized to achieve satisfactory performance.

4.2 Modified resolution synthesis

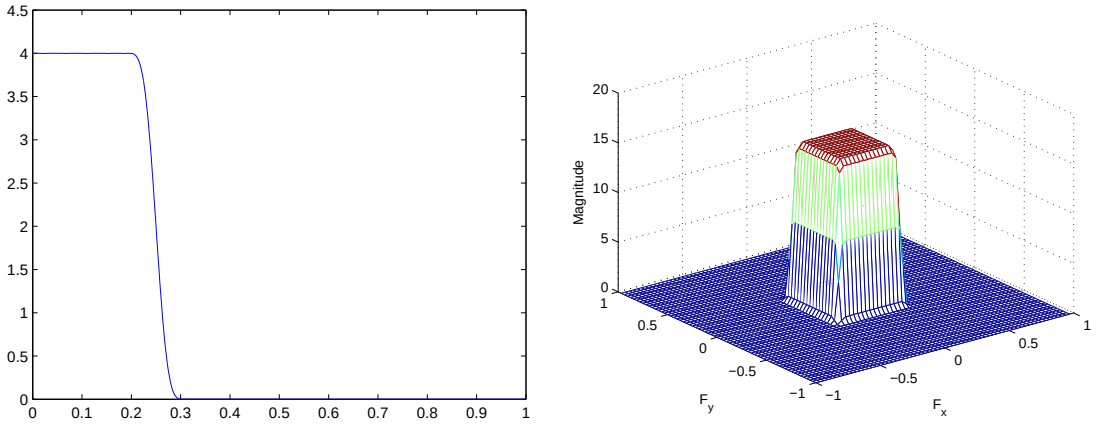
In its current form resolution synthesis is computationally too demanding for systems with limited computational resources and memory. The high computational load is mainly due to the large number of classes required for satisfactory performance (typically anywhere between 30-100), and the requirement for weighted linear combination (soft filtering). Soft filtering is especially demanding since it requires repeated application of a 5×5 filter, implying 25 additional multiplications, and an additional accumulation for every class included in soft filtering. In addition, the combination weights (w_j's in Figure 4.2 must be

this implies we have 3×3 filter phases. Note that this is even smaller than the bicubic filter, which is 4×4.



(a) (b) (c) (d)

Figure 4.4. (a) Original image (b) Bilinear interpolation result (c) 91×91 **separable filter result (d)** 12×12 **directional filter result**



(a) (b)

Figure 4.5. (a) Frequency response of the 91 tap 1D filter (b) 2D frequency response of the 91×91 **separable filter**

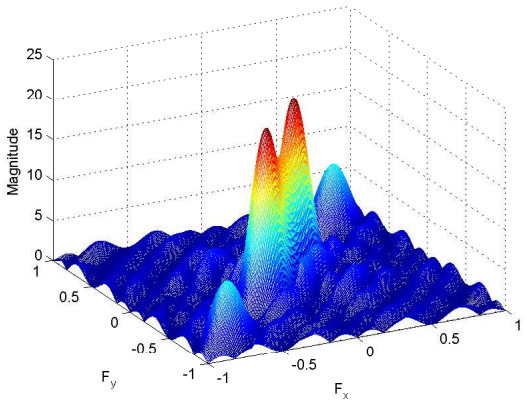


Figure 4.6. 2D frequency response of the 12×12 directional filter

computed to obtain the final result. Although in [51] Atkins proposes several modifications to reduce computational complexity, these modifications do not allow for efficient hardware implementations, and require large amounts of on-board memory. During our experiments to gauge the performance of resolution synthesis, we observed that directly reducing the number of classes (below ~ 30) without any modifications in the training, feature extraction or classification stages, severely degrades performance. Visual performance of the resolution synthesis algorithm trained with eleven context classes is demonstrated in Figures 4.7, 4.8 and 4.9. Figure 4.7 shows the original test image that is enlarged by a factor of two in both vertical and horizontal directions. Figures 4.8 and 4.9 compare regions cropped from the results of bicubic interpolation and original resolution synthesis trained with eleven classes and approximately 250,000 training (high-low) pixel pairs. The visual performance of resolution synthesis under current settings is equivalent to that of bicubic interpolation, which is a much simpler method. Also using only one class (the class with maximum membership) to compute the high resolution pixels resulted in degraded performance. We found out that the discrimination power of the feature vectors shown in Figure 4.2 was severely degraded as the number of context classes was reduced below ~ 25, which in turn degraded the performance of the training.

These shortcomings render resolution synthesis useless for customer grade flat panel

Figure 4.7. Original test image

displays, where the computational complexity must be kept below some threshold. *Our goal is to introduce novel modifications to allow resolution synthesis to perform satisfactorily with as low as eleven context classes using hard decision - i.e., using a single class for filtering*. Proposed modifications are in the feature extraction and classification blocks, and in the training method used to extract context classes, class prototypes and the optimal filters from the available training data.

4.2.1 Proposed modifications in feature extraction

Before we discuss the properties of the optimal feature vectors, we pause to elaborate on difficulties associated with defining and finding the *optimal* features. Apparently, the feature extraction mapping has a strong influence on the visual performance of the algorithm. The feature mapping along with the clustering algorithm employed during the off-line training phase effectively *define* the resulting context classes. Note that the clustering algorithm has a rather limited effect. The dominant structure of the feature vector space is mainly decided by the feature extraction mapping. The clustering algorithm simply detects and

(a) (b)

Figure 4.8. (a) Bicubic interpolation and (b) original resolution synthesis

(a) (b)

Figure 4.9. (a) Bicubic interpolation and (b) original resolution synthesis

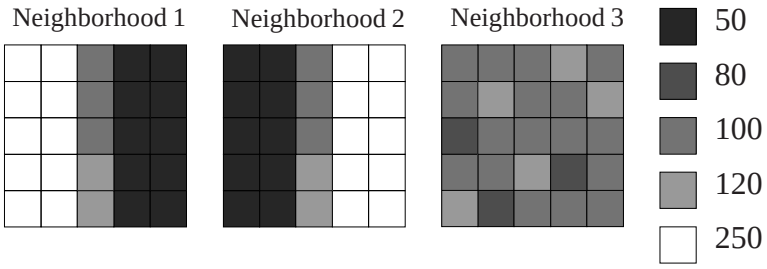


Figure 4.10. 5×5 **low resolution neighborhoods with varying spatial structures**

exploits this structure by grouping pixels with similar feature vectors together, and obtaining prototype feature vectors that best explain all pixels assigned to each class.

Let us start with discussing the notion of optimality for features. Although most of our algorithmic development and performance evaluation was based on mean squared error [6] (MSE) minimization, final evaluation of our resolution enhancement algorithm will be done subjectively by human observers. Hence, it makes sense to define the optimality of the feature set and the resulting resolution enhancement algorithm as *maximizing perceived visual quality for a fairly large set of natural image/video signals*. Unfortunately, understanding and modeling the way humans assess the quality of image/video signals is a challenging research problem. Although there exist subjective quality metrics based on experimental studies of the human visual system, the exact quality criteria employed by humans are still unknown. The subjective quality assessment task is further complicated by the fact that the way humans evaluate quality of image/video signals, in terms of sharpness, contrast, and visual artifacts, is dependent on the viewing conditions [52]. Certain visual artifacts such as color contouring become visible when viewed in dim light, while jagged edges are easier to see in high contrast scenes viewed in well lighted rooms. Finally, to make things even worse, perceived visual quality is a strong function of the viewer's personal taste. Some viewers enjoy sharp pictures with high contrast, and may be willing to accept certain visual artifacts commonly produced when obtaining such pictures (such as over and undershoots around edges). On the other hand, some viewers prefer artifact free pictures at the expense

[6] Since the low resolution images are obtained from high resolution originals, we have access to perfectly registered high resolution pixel values, and it is possible to compute MSE values.

of reduced sharpness.

All these observations lead to the conclusion that accurately measuring the subjective visual performance is an involved and time consuming process. Furthermore, reproducibility and/or generalization of the measurements is not guaranteed since the results are dependent on the viewing conditions and the selected viewers. In light of this fact, futility of searching for a single set of visually *optimal* feature set becomes obvious. Hence, we rather try to design our feature vectors to achieve simpler goals, namely, to provide good edge synthesis and robustness against signal noise. The properties we try to enforce on our feature vectors in the following are merely ways of achieving these goals within certain hardware limitations, and apparently we do not claim any optimality in the subjective visual quality sense.

Ideally, we require the feature vectors to be insensitive to changes in the average luminance value of the local neighborhood. This guarantees that blocks with similar structure but different average luminance, such as edges of the same orientation with different step size or polarity, will be clustered together. The feature mapping proposed by Atkins does not possess this property. For an illustration of this problem consider the 5×5 pixel neighborhoods shown in Figure 4.10. Judging by the dominant spatial structure, we expect to find the feature vectors extracted from the first two neighborhoods, which represent a vertical edge, to be closer to each other compared to the feature vector extracted from the third window, which represents a uniform region with a slight texture. Using the pixel values given in the Figure 4.10, we compute that the distance between the first and the second feature vectors is 58.15, the distance between the first and the third feature vector is 14.08, and finally the distance between the second and the third feature vectors is 31.52. Clearly, the distance between the feature vectors extracted from clear vertical edges (58.15) is larger than the distances between these feature vectors and the feature vector extracted from a uniform region (14.08 and 31.52). This shortcoming of the feature vectors results in creation

of redundant context classes, which model essentially the same feature with different luminance values. When we are allowed to have a sufficiently large number of classes, this does not severely effect the performance. However, if we have a very limited number of context classes, this problem has a devastating effect on the visual performance. Since certain spatial structures can have higher frequency in the training set, they dominate one or more other context classes with lower frequencies, and capture multiple classes. This severely degrades the visual performance since the pixels belonging to the suppressed classes are processed with filters effectively optimized for a different spatial structure. To improve the feature vectors we start with a list of desired properties

1. Feature vectors should not heavily depend on the average illuminance value of the local neighborhood.
2. Feature vectors of edges with the same orientation but different polarity should be close.
3. Feature vectors should effectively capture the dominant local spatial structure even under slight aliasing and/or noise.

To reduce the dependency on the local illuminance, the feature extraction block is modified to have a mean removal step. Right after the first order pixel differences are obtained, the mean value of the differences is computed and subtracted from all elements. To satisfy the requirement that edges with the same orientation with different polarity should have similar feature vectors, the next modification is to square all terms of the feature vector. We have also experimented with taking the absolute value of all terms, but using squared terms was found to have an additional advantage. When we raise all terms to the second power, larger differences representing the dominant structured are effectively boosted compared to the smaller differences that might have been caused by noisy or slightly textured pixels. Since the final step of feature extraction is normalization, boosting larger differences results in further suppressing smaller differences in the final feature vector. And finally, to increase robustness against noise and slight aliasing, we extract feature vectors from a 5×5

window instead of a 3×3 window. This increase in the window size also increases the discrimination power of the feature vectors, especially for edges that are close to the perfect vertical/horizontal orientations. Note that these edges are hardest to detect and synthesize properly. Since the angle of such an edge to perfect vertical/horizontal orientation is very small, the amount of change in values of pixels within a small window centered on the edge boundary is also very small. Hence, to effectively handle these edges we require an increase in the size of the window from which we extract the feature vectors. Features extracted from 5×5 neighborhoods with the modifications detailed so far proved to have much higher discrimination power, and performed much better under slight aliasing. All these modifications resulted in the improved feature vectors obtained as the normalized version of $\tilde{\boldsymbol{y}}$, *i.e.*,

$$y_i = \frac{\tilde{y}_i - \tilde{\mu}}{\left[\sqrt{\sum_{j=1}^{8}\left(\tilde{y}_j - \tilde{\mu}\right)^2}\right]^p} \quad \text{with } \tilde{\mu} = \frac{1}{20}\sum_{j=1}^{20}\tilde{y}_j \quad \text{and } p = 0.75, \tag{4.5}$$

where $\tilde{\boldsymbol{y}}$ is defined as in Figure 4.11.

The original resolution synthesis algorithm models the feature vectors as samples drawn from a multivariate Gaussian mixture. The mixture parameters are estimated from the training data via the EM algorithm. The resulting model is used to compute the probability of a feature vector being drawn from a given mixture class. These probabilities are interpreted as a fuzzy clustering of the feature vectors. Instead of EM based clustering we use more flexible robust fuzzy-clustering techniques. Since fuzzy-clustering techniques are based on the minimization of a cost function, various constraints on cluster sizes, membership degrees and noisy samples can be imposed by modifying the cost function. As elegant as the EM based clustering may be, such constraints can not be imposed in the Gaussian mixture framework. The fuzzy clustering technique we use is a generalization of the well-known Gustafson-Kessel algorithm. We combine the Mahalanobis distance with cluster volume constraints such that the cluster corresponding to the uniform areas has a larger volume

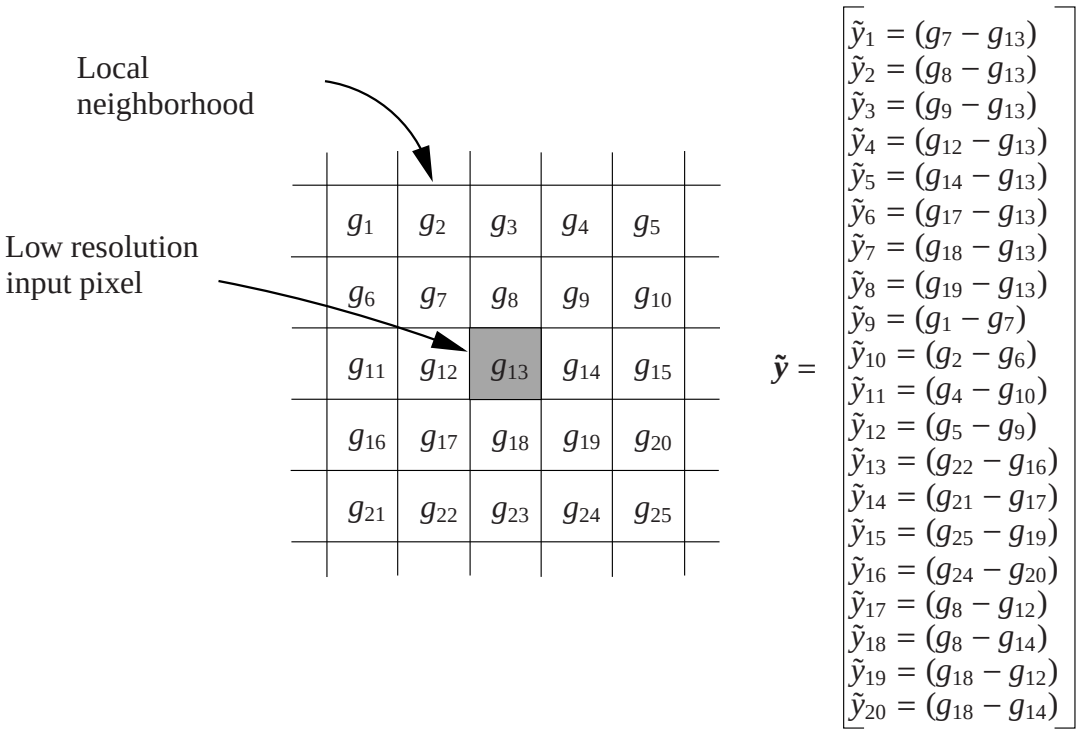


Figure 4.11. 5×5 **low resolution neighborhood and the modified feature vector**

compared to the other clusters. This constraint is based on the observation that natural images mainly consist of uniform areas. Even edges have uniform areas around the transition boundary that defines the edge.

The reduction in the number of context classes requires the classification step to be improved accordingly. Now that we have a much smaller number of context classes, faulty classifications seriously degrade the overall performance. An obvious shortcoming of resolution synthesis training phase is the use of a single standard deviation for all feature coordinates of all classes. This simply corresponds to using $\boldsymbol{C} = \sigma^2 \boldsymbol{I}$ as the covariance matrix of all classes where $\boldsymbol{I}$ is the 8×8 identity matrix. It should be self-evident that the feature vectors extracted from neighborhoods that belong to different context classes may have quite different magnitudes (energies or variations). Hence imposing all classes to have the same covariance matrix is simply poor modeling. We remedy his shortcoming by using different a variance value for each class, *i.e.*, $\boldsymbol{C}_i = \sigma_i^2 \boldsymbol{I}$. This corresponds to assuming that the entries of the feature vectors are identical and independent, and the variances of the

feature vector elements are class dependent.

The objective of our proposed research is to render the proposed single-frame resolution enhancement algorithm applicable to customer grade flat panel displays. In its current form the proposed scaling algorithm is computationally simple enough to be implemented in such systems. But computational simplicity by itself is not enough to certify an algorithm fit for customer grade products. There are other factors that influence the applicability of any algorithm aimed for these products, the most important one being robustness under varying operating conditions, and a wide range of input signals. To make the proposed method suitable for TV sets, we require robustness against noise and specific visual artifacts frequently observed in TV broadcasting, namely, deinterlacing artifacts.

As detailed in Section 4.1, the proposed scaling method classifies the input pixels based on local neighborhoods. If the signal to noise ratio of the input signal is low enough, noisy pixels may introduce classification errors. In typical TV video pipelines denoising is performed by a dedicated block that is placed right after the video decoder. In our work, we assume that some type of denoise filtering is performed on the input signal, and the SNR of the signal input to the scaler is not very low.

Since signal noise is typically of high frequency nature, noisy pixels are erroneously classified as genuine details and *enhanced* by resolution synthesis, resulting in a visually disturbing, noisy appearance. Directly applying resolution enhancement to noisy input signals creates two types of visual artifacts. First is the flickering pixels caused by temporally changing classification results for the noisy pixels. Second is the grainy look caused by amplified static noisy pixels. To suppress these artifacts we propose two modifications in the feature extraction block. The first modification is to include a coring function right after pixel differences are computed. The coring function is defined as

$$d_{out} = \begin{cases} 0 & \text{if } d_{in} \leq T_l \\ d_{in} & \text{if } T_l \leq d_{in} \leq T_h \\ 0 & \text{if } T_h \leq d_{in} \end{cases} \tag{4.6}$$

where d_{in} is the input pixel difference value, d_{out} is the output pixel difference value, and T_l, T_h are the lower and higher thresholds, respectively. In our implementation the thresholds T_l and T_h are fixed values optimized for certain noise levels (we used $T_l = 4$ and $T_h = 256$). If the computational budget allows, the thresholds can be made adaptive to the local contrast and signal dependent noise levels.

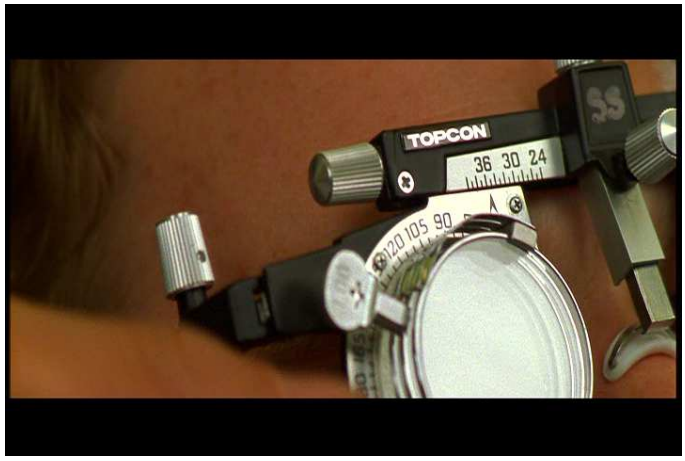
The second modification is detection and suppression of input pixels with very high feature values. Based on the observation that noisy pixels typically have high gradients in multiple directions, we detect and suppress pixels whose features satisfy the following conditions

$$\text{Condition 1: If } \tilde{y}_1 > T_a \text{ and } \tilde{y}_3 > T_d \text{ and } \tilde{y}_6 > T_d \text{ and } \tilde{y}_8 > T_a \tag{4.7}$$

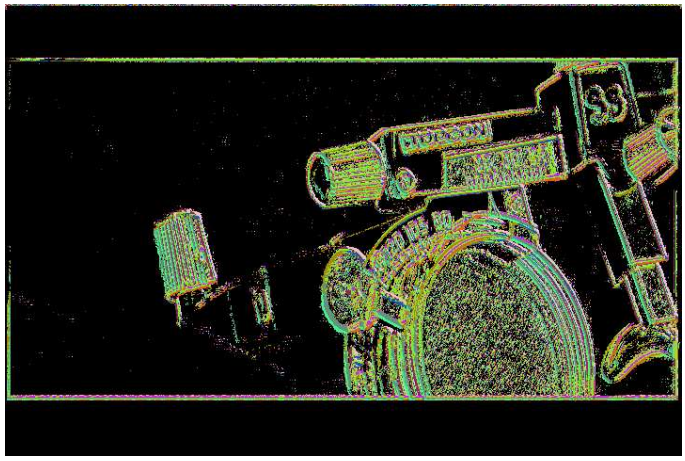
$$\text{Condition 2: If } \tilde{y}_2 > T_v \text{ and } \tilde{y}_4 > T_h \text{ and } \tilde{y}_5 > T_h \text{ and } \tilde{y}_7 > T_v \tag{4.8}$$

where, the unnormalized feature vector entries $\tilde{y}_i$ are as defined in Figure 4.11. Figure 4.12(a) shows a frame captured from a DVD. Classification results with and without coring are shown in Figure 4.12(b) and (c), respectively. Every color corresponds to a specific context class. Reduced business in uniform areas implies that a much larger portion of the noisy pixels are classified as uniform, denoted by black color. In our implementation the thresholds T_h, T_v, T_d and T_a are fixed values optimized for certain noise levels (we used $T_h = T_v = T_d = T_a = 40$). If the computational budget allows, these thresholds can be made adaptive to the local contrast.

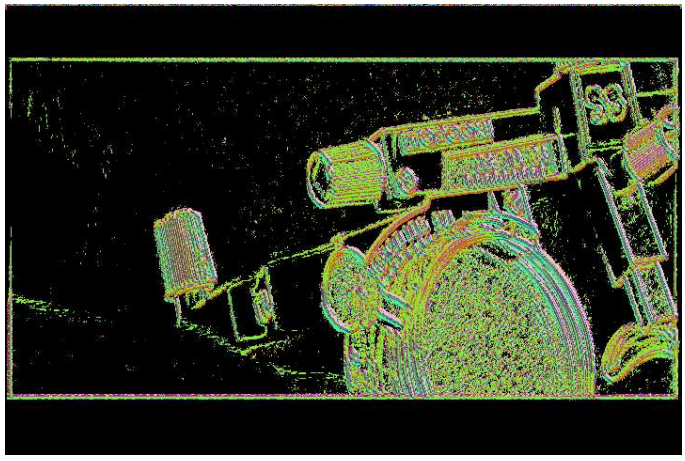
Another important perturbation in the signal structure is interlacing. NTSC standard for conventional color TV broadcasting allocates a fixed bandwidth of 6 MHz for each TV channel. Since the allocated bandwidth is limited, the amount of information that can be conveyed through the channel is also limited. Noting that information in this case is a *video* signal, one can see that the main trade-off is between the spatial resolution (quality and amount of details in the spatial domain, *i.e.* how densely we sample in the spatial domain) and the temporal resolution (smoothness of the captured motion, *i.e.*, how densely we



(a)



(b)



(c)

Figure 4.12. (a) The original captured frame without any modifications including addition of Gaussian noise (b) Classification result obtained *with* coring (c) Classification result obtained *without* coring

sample in the temporal domain). Hence, the question that naturally arises is: *How should we utilize the bandwidth to provide the viewer with satisfactory spatial and temporal resolution while meeting the bandwidth constraint?* Interlacing provides a means of trading (vertical) spatial resolution for better temporal resolution. Instead of transmitting all pixels of every frame, only the odd or even numbered lines of the frames (usually called as odd and even fields) are captured, and transmitted in an alternating fashion. The TV hardware is responsible for combining the odd and even fields, and displaying the full frame. The CRT TV sets achieve this by first scanning the phosphor screen with the odd field first, and introducing a single line shift before scanning with the even field. The afterglow of the phosphor screen, combined with the persistence of vision results in two fields being perceived as a single continuous image. The result is viewing full horizontal detail with half the bandwidth that would be required for a full progressive scan, while keeping the necessary CRT refresh rate to prevent heavy flicker. Unfortunately, flat panels displays, including LCDs and plasma displays, are inherently progressive scan, that is, they can not directly display interlaced video signals. Depending on the design of the display, deinterlacing is performed either within the video pipeline or before the video signal is input to the display. In either case deinterlacing can get quite involved. The main issue is that the boundaries of moving objects are displaced in odd and even frames, and without precise knowledge of the motion present in the scene perfect reconstruction is not possible. Motion compensated deinterlacing techniques are computationally too intensive for low and mid-end products, and typically these products have much simpler deinterlacers based on line and/or field repetition or averaging. These algorithms inevitably create visual artifacts on and around object boundaries in high motion scenes, namely blurred object boundaries (line repetition or averaging) and comb artifacts (caused by field repetition).When processed with single frame resolution enhancement algorithms, these artifacts are erroneously considered as genuine image details, and amplified, resulting in visually disturbing pictures.

To remedy this issue we propose a feature based approach similar to the one used for

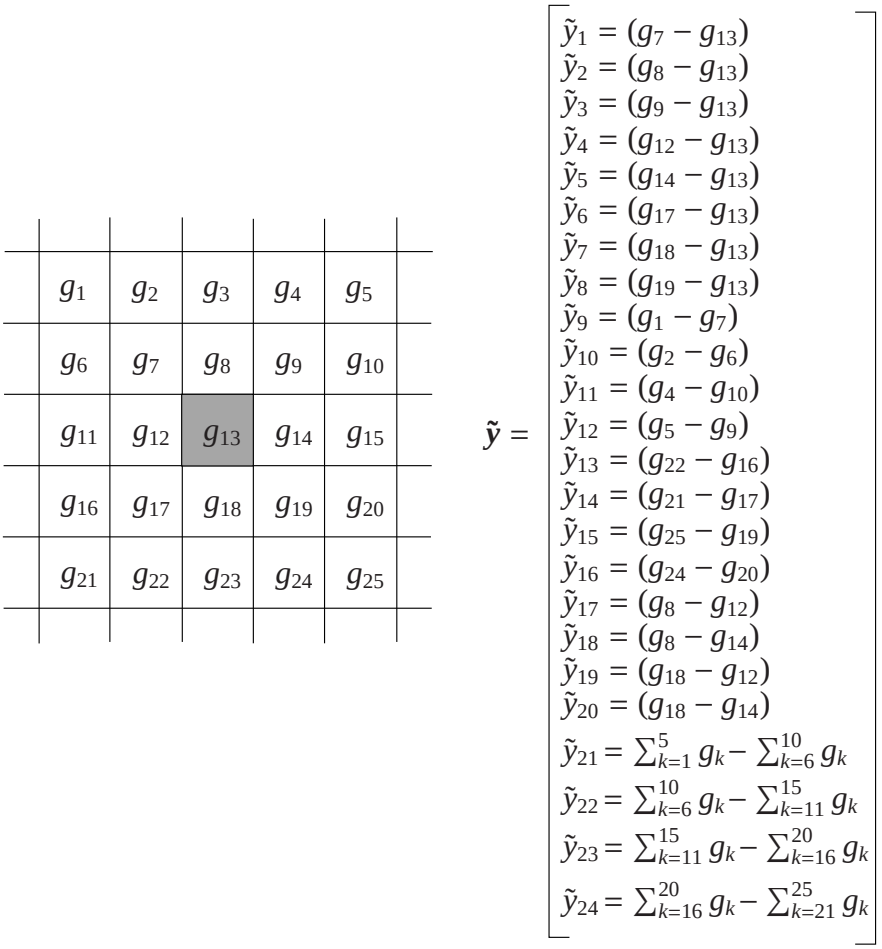


Figure 4.13. 5×5 **low resolution neighborhood and the modified feature vector**

noisy signals. Our approach is based on the observation that deinterlacing artifacts exhibit a very specific spatial structure of alternating horizontal lines as shown in Figure 4.14(a). We augment feature vectors with four additional features designed to detect this spatial structure. The resulting modified feature vector is shown in Figure 4.13. If the values of the augmented features satisfy the condition given as

$$\tilde{y}_{22} > T_{sng} \text{ and } \tilde{y}_{23} > T_{sng} \text{ and } \tilde{y}_{21} > T_{dbl} \text{ and } \tilde{y}_{24} > T_{dbl}, \tag{4.9}$$

then the input pixel is declared as deinterlacing artifact, and interpolated with a linear filter that does not amplify the disturbing line structure. Figure 4.14(b) shows the deinterlacing artifacts detected in the example deinterlaced picture shown in Figure 4.14(a). Once again, in our implementation the thresholds T_{sng} and T_{dbl} are fixed values optimized over a set

(a)

(b)

Figure 4.14. (a) Deinterlaced frame with visible line artifacts (b) Deinterlacing artifact map

video clips (we used T_{sng}=60 and T_{dbl}=60). If the computational budget allows, these thresholds can be made adaptive to the local motion and contrast.

4.2.2 Iterative training scheme

In this section we discuss the details of our final algorithmic modification, namely, feature-filter coupled iterative training. The proposed training method, shown in Figure 4.15, is based on the observation that interpolation filter design stage has direct access to the high resolution pixels. We note that due to Assumption 3 given in Section 4.1 clustering with respect to feature vectors (distribution parameter estimation) is completely uncoupled with

filter design, and high resolution pixels are only utilized by the filter design block, which is executed only once after the pixel clustering block converges. Hence, if we can couple interpolation filters to the feature extraction and classification stages, clustering performance should be improved by the contribution of the high resolution spatial information. Given the low and high resolution training images, proposed method iteratively extracts the best interpolation filters and the context class prototypes that are used to determine input pixel's context. The iterative training works as follows.

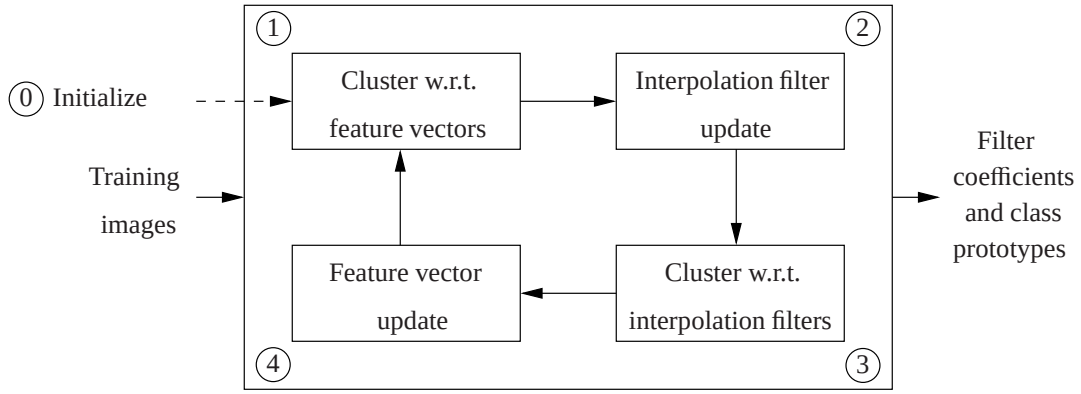


Figure 4.15. Proposed iterative training scheme

0. ***Initialization***

   After extracting the feature vectors of all the low resolution pixels in the training set, class prototypes are initialized randomly. The prototype for class number one is manually set to a vector of all zeros. This guarantees that we have a class reserved for uniform areas. All covariance matrices are set to identity matrices.

1. ***Clustering with respect to features***

   After initialization, the low resolution pixels are classified with respect to their feature vectors, Block 1 in Figure 4.15. This is done by going through all low resolution pixels, computing the weighted Euclidian distance (the weighting matrix is the inverse of feature covariance matrix) between the pixel's feature vector, which is a representative of the local image characteristic of the low resolution pixel and the cluster prototypes, which are representatives of different context classes. Then the

input low resolution pixel is labeled with the index of the cluster whose feature vector is the closest to the low resolution pixel's feature vector.[7]

2. ***Filter update***

   Once the low resolution pixels are clustered with respect to their feature vectors, *i.e.* spatial context, the interpolation filters for all clusters are updated with the filter that minimizes the mean-squared-error between the interpolated and the true high resolution pixels computed for all low resolution pixels in a specific cluster, Block 2 in Figure 4.15. Tikhonov regularization is used to avoid filters that excessively amplify the high frequency components. While preparing the training samples, a small amount of blurring prior to downsampling (anti-alias filtering) is necessary to model the camera response and also to avoid aliasing. But completely filtering out the high frequency components effectively creates an inverse problem where the filters are asked to bring back completely removed signal components (this is only possible in multi-frame case), resulting in bad filters. Hence, the design of the anti-alias filter is quite important. We observed that zero phase linear filters with properly adjusted cut-off frequencies provided good visual results.

3. ***Clustering with respect to filters***

   After the filter update, all input pixels are clustered with respect to the minimum mean-squared-error interpolation filter, Block 3 in Figure 4.15. This is accomplished by going through all low-resolution training pixels, computing the interpolated high resolution pixels by all interpolation filters one by one, and comparing the interpolated pixels to the available high resolution pixels. The low resolution pixel is then labeled with the index of the interpolation filter that gives the minimum mean-squared-error between the interpolated and real high resolution pixels.

[7] Note that any clustering scheme can be used in this stage. We experimented with EM based Gaussian mixture fitting, but we did not observe any visual quality improvement.

4. ***Class prototype update***

   Once all the input pixels are classified, the feature vectors of the obtained clusters are updated one by one, Block 4 in Figure 4.15. This update can be done in various different ways such as taking the average of the median of the feature vectors. Class covariance matrices are updated next. To reduce computational complexity, we assume diagonal covariance matrices. Then we go back to clustering with respect to features, and iterate in this fashion for predetermined times. In our experiments we worked with 2 iterations.

Once the filter coefficients and the feature vectors of all contexts are learned from training data, these parameters are passed to the interpolation stage. We note that further improvements over the algorithm detailed here are possible. It should be clear to the reader that clustering with respect to the filters and clustering with respect to the features are two different goals which may not agree for a specific choice of features and filter clustering method. Although it is possible to come up with a feature extraction method and a way of clustering pixels with respect to the best filter that agree for an arbitrarily large percentage of training pixels, finding such schemes is not straightforward. We have observed that for the current implementation increasing the number of iterations corrupted the interpolation filters and the class prototypes. Through exhaustive computer simulations we have concluded that clustering with respect to interpolation filters based on minimum MSE is the main reason that avoids convergence. Pixels in uniform areas are frequently assigned to wrong context classes due to their lack of structure (almost all filters perform good). Additional regularization terms are required to make clustering with respect to filters more robust.

One aspect of the algorithm which we should discuss is the number of context classes. Obviously, too few context classes would not be able to represent all spatial structures frequently observed in natural image/video data. On the other hand, too many classes would over-fit to the training data, degrading the general interpolation performance. In the

original resolution synthesis derivation the suggested number of context classes to obtain satisfactory visual results is around M=100. In [51] it is argued that the number of context classes is a parameter that can be identified along with the other parameters of the model, under the a modified criterion of optimality called penalized[8] likelihood criteria [50]. In our case we were limited by the maximum allowable computational budget which was dictated by the Xilinx Spartan 3 FPGA board. Based on our study of the mentioned board we decided to limit our context classes with 13. All of the results presented in this thesis are obtained with using less than 13 context classes.

4.3 Extension to non-integer scaling ratios

The original resolution synthesis method is designed for integer scaling ratios only. This limitation is a serious drawback since most applications require non-integer scaling ratios. In this section we describe generalization of the resolution synthesis method to non-integer scaling ratios.

Let us denote the scaling ratios in horizontal (x) and vertical (y) directions with L_x and L_y, respectively. For integer scaling ratios larger than or equal to one, for every input pixel we have an $L_x \times L_y$ block of corresponding output pixels. When the scaling ratios are smaller than one, we have the downsampling case [9]. If we downsample by an integer ratio, a similar statement holds: For every $L_x \times L_y$ block of input pixels, there is one corresponding output pixel. For non-integer scaling ratios these statements are not valid. Since the filter design stage of the original resolution synthesis is based on this *pixel to patch* structure, non-integer scaling ratios can not be directly handled.

The registration of input and output pixel grids for non-integer scaling ratios is essential to understand the proposed modification. The key to understand non-integer scaling

[8] Each of these criteria includes a term for the likelihood of the model, which is weighed against a term for the order of the model, so that models with excessively low or high orders are discouraged.

[9] Typically decimation by L refers to keeping one out of every L samples and dropping the rest. Downsampling refers to anti-alias filtering followed by decimation. We use decimation and downsampling interchangeably assuming proper anti-alias filtering is applied.

ratios is the idea of *polyphase filtering* [53]. Instead of including a through overview of polyphase filtering, we will take a more practical approach to introduce the idea. For a detailed treatment of polyphase filtering the interested reader is referred to [53]. Also for an interesting application of polyphase filtering to image scaling and analysis of aliasing artifacts see [54]. For the sake of clarity, we shall introduce the basic idea on one dimensional (1D) signals. All the following results and ideas can easily be generalized to the two dimensional (2D) case.

The main result of polyphase filtering is the following: *Non-integer scaling ratios are handled by first upsampling by an integer factor U, followed by downsampling by an integer factor D, such that $\frac{U}{D}$ equals the desired scaling ratio L.* The order of upsampling and downsampling operations can be changed, but for upscaling, this order is the natural choice. Note that for the upscaling case $L \geq 1$, which implies $U \geq D$. Hence, by upsampling first we effectively eliminate the need for the anti-alias filtering stage that is required to avoid aliasing. We next introduce the idea of *filter phases*. Phases of an interpolation filter can be best described on an example. Consider Figure 4.16 where we demonstrate upscaling of a 1D signal ($x[n]$) by three. We use a nine tap LSI interpolation filter ($c[n]$). We first insert $L-1=2$ zeros between each sample, and then filter the resulting upsampled signal with the interpolation filter to obtain the upscaled signal $y[n]$. From Figure 4.16 we can see that for any given output sample, only three filter taps are multiplied with non-zero samples. The remaining filter taps fall on zeros inserted between the input samples, and do not contribute to the output. We can also see that the filter coefficients that correspond to non-zero input samples can be grouped into three *phases*

$$\text{Phase } 0 \;= \{c[1], c[4], c[7]\}, \tag{4.10}$$

$$\text{Phase } 1 \;= \{c[0], c[3], c[6]\}, \tag{4.11}$$

$$\text{Phase } 2 \;= \{c[2], c[5], c[8]\}. \tag{4.12}$$

Note that the number of filter phases is $L = 3$. One of the most important results from

polyphase filtering is the idea of filter phases. We emphasize that *for any output sample only a single phase of the interpolation filter is used*. The remaining phases overlap with zero samples. Hence, we can improve computational efficiency of interpolation by simply separating an interpolation filter into its phases, deciding which phase is to be used for an output sample, and performing the filtering with the appropriate phase only. If the desired scaling ratio was $L = 1.5$, we would continue with decimating by two. Since we upsample by three prior to decimation, anti-alias filtering is not necessary and the final scaled signal can easily by obtained by picking one out of every two samples.

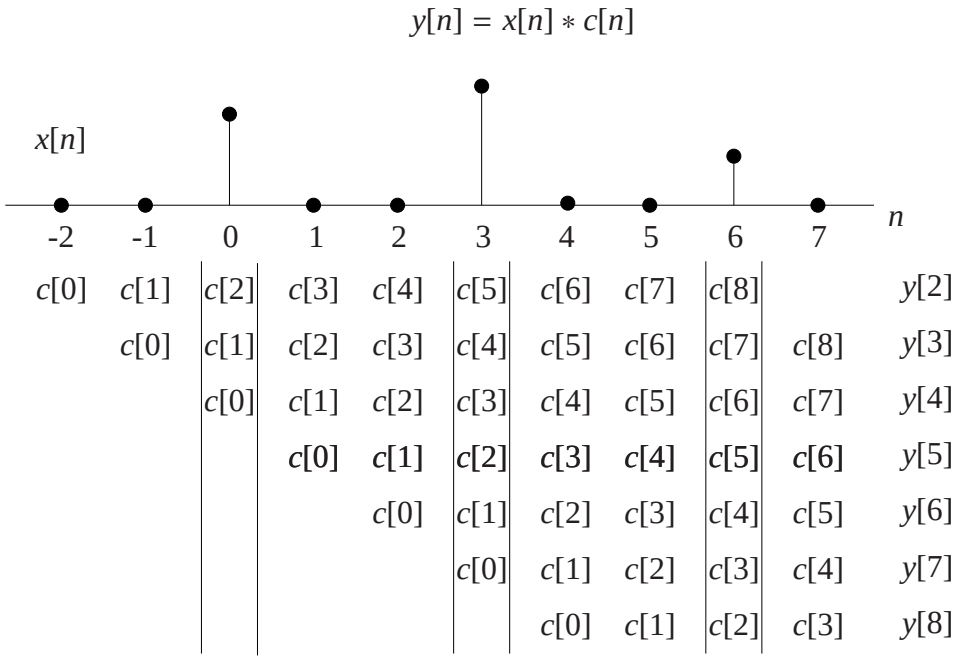


Figure 4.16. Phases of an interpolation filter

Figure 4.17 demonstrates two 2D upscaling examples for $L = 2$ and $L = 1.5$. The coloring of the pixels is designed to denote the pixel registration between the input and output patches. Every output pixel is also labeled with the vertical and horizontal filter phases. For non-integer scaling ratios, the number of phases is basically equivalent to U.

Hence,

$$\text{For } L = 2 = \frac{2}{1} \Rightarrow \text{ 2 phases} \tag{4.13}$$

$$\text{For } L = 1.5 = \frac{3}{2} \Rightarrow \text{ 3 phases} \tag{4.14}$$

For $L = 1.5$, the blue input pixel (which can be though as the $(0, 0)$ position on the low resolution grid) has a corresponding 2×2 patch of output pixels. Similarly, green and red pixels are mapped to 2 pixels and the yellow pixel gets mapped to a single pixel. It can be verified that this block structure is repeated for all other input pixels. The period of the repeating output block structure is $U = 3$. Hence, if the desired scaling ratio was $L = 1.6 = \frac{8}{5}$, we would have $U = 8$ phases, and the output patch pattern would repeat with every 8×8 block.

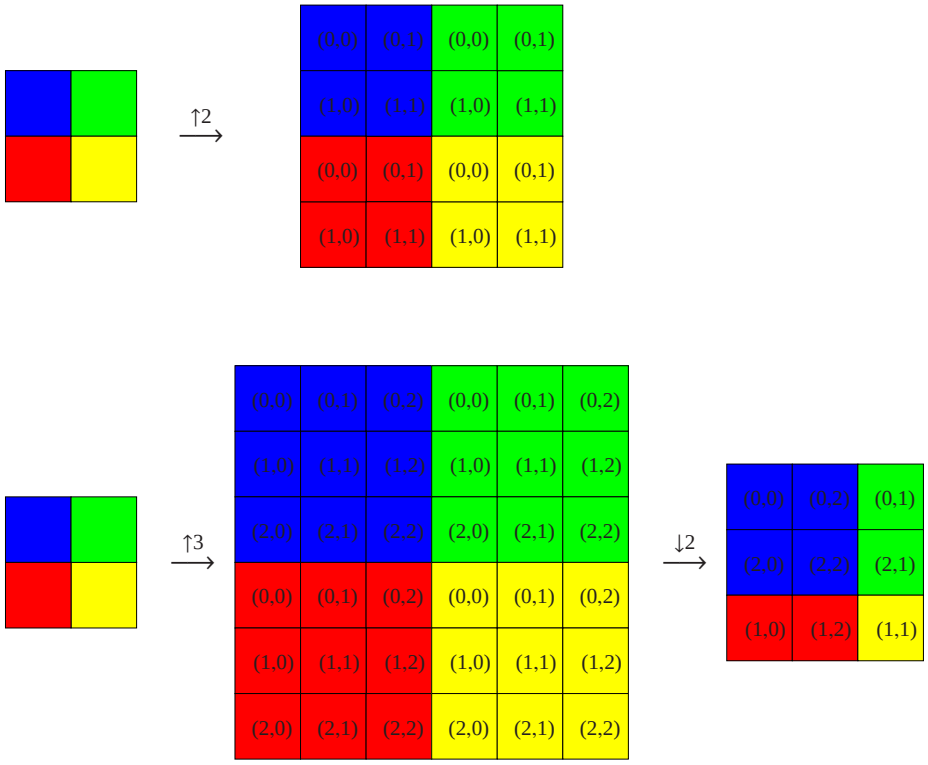


Figure 4.17. 2D scaling examples with $L = 2$ and $L = 1.5$

The next polyphase filtering idea is to perform filtering selectively to avoid wasting computational resources on the decimated samples. For the $L = 1.5$ case in Figure 4.17 we

can see that after decimation only one forth of the pixels in the intermediate picture make it to the final output. We could further improve computational efficiency by computing only the samples that will not be dropped by the decimation step. The implementation of this idea, which completes the polyphase scaling concept, is straight forward. For every output pixel we first compute the corresponding phase. It should be fairly clear that the phase of each output sample is directly decided by its location on the output grid. If we denote the output vertical and horizontal coordinates with (x, y), and the vertical and horizontal phases with p_x and p_y, one can easily verify that

$$p_x = \left\lfloor L_x \times \left(\frac{x}{L_x} - \left\lfloor \frac{x}{L_x} \right\rfloor \right) \right\rfloor,$$
$$p_y = \left\lfloor L_y \times \left(\frac{y}{L_y} - \left\lfloor \frac{y}{L_y} \right\rfloor \right) \right\rfloor.$$

Once we decide on the phase number, we apply the corresponding interpolation filter phase to obtain the output pixel value. This implementation combines phase based filtering with decimation to achieve computationally efficiency non-integer ratio scaling.

From a practical point of view the polyphase filtering strategy is not directly applicable to hardware implementation because of two reasons:

1. In theory, we can always get arbitrarily close to the desired L (as long as L is a rational number) by properly adjusting U and D. The problem is, for some values of L, the integers U and D can get too large. Fortunately, polyphase filtering theory offers elegant solutions for large to this problem. Note that direct interpolation approach would use all filter coefficients and store the upsampled image prior to decimation. For large values of U, we require large interpolation filters to achieve satisfactory visual results and the computational burden of filtering is substantial. Also, storing the upsampled image would cause memory issues since the upsampled image has U^2 times as many pixels as the input image. In actuality, thanks to polyphase filtering, we never execute full upsampling, store the upsampled image and then decimate. Nevertheless, in certain cases, getting an exact match requires U values so large that

even with polyphase filtering techniques the computational burden is prohibitive.

2. In theory, this approach can handle any non-integer valued scaling ratio but picking proper values of U and D. But in practice we do not have the luxury of changing these values for every scaling ratio. In actuality, we have control over only D, and we are stuck with a fixed value of U. This is due to the fact that every different value of U requires a new interpolation filter, and storing the corresponding filter coefficients. Since typical hardware flows are highly optimized, this type of flexibility is not possible.

Hence, the bottom line is *most of the time we have to settle for an approximate L*. Real world scalers handle non-integer scaling ratios by upsampling with a quite large ratio, typically around 128, followed by decimation, where the decimation ratio D is picked to match the desired *L as closely as possible*. Note that the larger the value of the fixed upsampling ratio, the better the approximation but the harder the interpolation filtering. Using polyphase filtering techniques, the interpolation filter is divided into its phases, and each output sample is computed by using a single phase only. This substantially reduces the computational burden of the interpolation filter. Also, by performing filtering for each output sample separately the storage problem is virtually eliminated.

We could directly apply the polyphase filtering approach (with fixed U and variable D) to generalize resolution synthesis to non-integer scaling ratios. But this approach is plagued by practical issues. Assuming we pick the fixed upsampling ratio as 128, we would have $128 \times 128 = 16384$ phases for each class. Each phase is an $M \times M$ non-separable filter, where typical values for M are 3,4,5. Optimizing such a large number of filter coefficients (for $M = 5$, we have 409600 coefficients for *each* class) is problematic. To obtain reliable filter coefficients we have to avoid ill-conditioned cases, where the number of linearly independent training samples for a phase is less than the data. Given the number of coefficients to be optimized, avoiding the ill-conditioned cases requires prohibitive amounts of data. Hence, we decided to take an alternative approach motivated by the observation that for

TV sets, high quality resolution conversion is only required for certain fixed ratios. This is the case since there exists only a limited number of native screen resolutions and a limited number of standard signal resolutions. Based on this fact, *we design a specific set of RS filters for each scaling ratio*. We observed that the visual quality improvement gained by using class specific non-separable filters allows us to lower the number of phases compared to LSI filtering.

We conclude this section by presenting a discussion of the optimality of the proposed generalization strategy. We base our discussion on a reformulation of the original optimal filter derivation presented in [51]. Let us start with definitions. The number of pixels in the j^{th} context class is denoted by N_j and defined as

$$N_j = \sum_{s \in S} w_{j,\boldsymbol{y}_s}, \tag{4.15}$$

where s is the index over the training set S. For the fuzzy clustering case the class weights $0 < w_{j,y_s} < 1$ are computed by the clustering method. For example, if we use Gaussian mixture fitting as done in the original resolution synthesis formulation

$$w_{j,\boldsymbol{y}_s} = p_{J|Y}(j|\boldsymbol{y}_s, \hat{\theta}), \tag{4.16}$$

where $\hat{\theta}$ denotes the Gaussian mixture parameters. For the crisp clustering case the class weights will be zero for all classes except for the class that the pixel is assigned to. The augmented mean vectors $\boldsymbol{\nu}_j$ are defined as

$$\begin{aligned} \boldsymbol{\nu}_j &= \begin{bmatrix} \boldsymbol{\nu}_{f|j} \\ \boldsymbol{\nu}_{g|j} \end{bmatrix} \\ &= \frac{1}{N_j} \sum_{s \in S} \begin{bmatrix} \boldsymbol{f}_s \\ \boldsymbol{g}_{s.} \end{bmatrix} p_{J|Y}(j|\boldsymbol{y}_s, \hat{\theta}). \end{aligned} \tag{4.17}$$

And finally the augmented class covariance matrices are defined as

$$\begin{aligned}\boldsymbol{\Gamma}_j &= \begin{bmatrix} \boldsymbol{\Gamma}_{ff|j} & \boldsymbol{\Gamma}_{fg|j} \\ \boldsymbol{\Gamma}_{fg|j} & \boldsymbol{\Gamma}_{gg|j} \end{bmatrix} \\ &= \frac{1}{N_j} \sum_{s \in S} \begin{bmatrix} (\boldsymbol{f}_s - \boldsymbol{\nu}_{f|j}) \\ (\boldsymbol{g}_s - \boldsymbol{\nu}_{g|j}) \end{bmatrix} \begin{bmatrix} (\boldsymbol{f}_s - \boldsymbol{\nu}_{f|j})^T & (\boldsymbol{g}_s - \boldsymbol{\nu}_{g|j})^T \end{bmatrix} p_{J|Y}(j|\boldsymbol{y}_s, \hat{\theta}). \end{aligned} \tag{4.18}$$

Using these definitions the optimal filter coefficients are computed as

$$\begin{aligned}\boldsymbol{A}_j &= \boldsymbol{\Gamma}_{fg|j} \boldsymbol{\Gamma}_{gg|j}^{-1}, \\ \boldsymbol{\beta}_j &= \boldsymbol{\nu}_{f|j} - \boldsymbol{\Gamma}_{fg|j} \boldsymbol{\Gamma}_{gg|j}^{-1} \boldsymbol{\nu}_{g|j}. \end{aligned} \tag{4.19}$$

In the original derivation presented in [51] the feature and filter parameters are obtained as ML estimates from the incomplete data likelihood using expectation maximization. Let us revisit Assumption 3 stated in Section 4.1:

Assumption 3: Given feature vector $\boldsymbol{y}$, the class distribution is independent of the high resolution and low resolution pixels

$$p_{J|F,G}(j|\boldsymbol{f}, \boldsymbol{g}) = p_{J|Y}(j|\boldsymbol{y}).$$

Using this assumption, the incomplete data likelihood can be factored into 2 terms that can be optimized separately. First, the feature parameters $\hat{\theta}$ are obtained by applying the expectation-maximization (EM) method. Then using the obtained ML estimates, the filter term is maximized. The derivation of the optimal interpolation filters is mainly based on Assumption 2 stated in Section 4.1:

Assumption 2: Given the input low resolution pixel neighborhood and the context class, the high resolution pixels are Gaussian

$$p_{F|G,J}(\boldsymbol{f}|\boldsymbol{g}, j) = \mathcal{N}(\boldsymbol{A}_j \boldsymbol{g} + \boldsymbol{\beta}_j, \sigma^2 \boldsymbol{A}_j^T \boldsymbol{A}_j).$$

Hence, filter optimization is equivalent to ML estimation of the mean of a Gaussian. Since the filter coefficients are computed as the ML estimate of the mean of a Gaussian, the

resulting estimates are equivalent to minimum mean squared error (MMSE) estimates. Furthermore, for Guassian distribution MMSE of the mean is equivalent to the linear MMSE (LMMSE) [55]. The resulting formulas confirm this fact since they exactly match the well-known LMMSE solutions. Based on this observation we present an alternative derivation which is more suitable for our purposes.

Let us formulate the optimal filtering operation as

$$\boldsymbol{f}_s = \boldsymbol{A}_j \boldsymbol{g}_s + \boldsymbol{\beta}_j, \tag{4.20}$$

where $\boldsymbol{f}_s$ denotes a raster-scanned vector of output pixel and $\boldsymbol{g}_s$ denotes a raster-scanned vector of input pixels that fall into the filter mask. Note that for non-integer scaling ratios the size of the output patch, hence the size of $\boldsymbol{f}_s$ depends on s. This does not really affect the result, but to avoid notational confusion we *define* $\boldsymbol{f}_s$ as a vector with fixed length of $L_x \times L_y$, where the unused phase entries are simply zeros. We define the cost function for the j^{th} class as

$$\sum_{s \in S} w_{j,\boldsymbol{y}_s} \| \boldsymbol{f}_s - \boldsymbol{A}_j \boldsymbol{g}_s + \boldsymbol{\beta}_j \|^2. \tag{4.21}$$

Noting $w_{j,\boldsymbol{y}_s}$ denotes the previously defined class membership weights, we can see that the cost function in Eq. 4.21 is the accumulated weighted squared error for the j^{th} context class. Finally, we define the filter optimization problem for the j^{th} context class as

$$\min_{\boldsymbol{A}_j,\, \boldsymbol{\beta}_j} \left(\sum_{s \in S} w_{j,\boldsymbol{y}_s} \| \boldsymbol{f}_s - \boldsymbol{A}_j \boldsymbol{g}_s + \boldsymbol{\beta}_j \|^2 \right). \tag{4.22}$$

In appendix App. B we show that the resulting optimal filter coefficients are identical to the ones in Eq. 4.19. Hence, these two formulations define the same problem. Let us decompose the squared error term and rewrite Eq.4.21 as

$$\sum_{s \in S} \sum_{k=0}^{L_x \times L_y} w_{j,\boldsymbol{y}_s} \left(f_{s,k} - \boldsymbol{a}_{j,k} \boldsymbol{g}_s + \beta_{j,k} \right)^2, \tag{4.23}$$

where $f_{s,k}$ is the k^{th} output pixel, $\boldsymbol{a}_{j,k}$ is the corresponding filter phase, and $\beta_{j,k}$ is the corresponding bias term. From Eq. 4.23 we can see that the cost function, hence the optimal

filter, can be decomposed into $L_x \times L_y$ different terms corresponding to each phase. This observation is the key to generalize the algorithm to non-integer scaling ratios. Basically, we train each phase of each class separately, i.e., we obtain the optimal filter phases and bias terms as

$$\min_{\boldsymbol{a}_{j,k},\, \beta_{j,k}} \left(\sum_{s \in S} w_{j,\boldsymbol{y}_s} \| f_{s,k} - \boldsymbol{a}_{j,k} \boldsymbol{g}_s + \beta_{j,k} \|^2 \right) \text{ for } k = 0, 1, \ldots, L_x \times L_y. \tag{4.24}$$

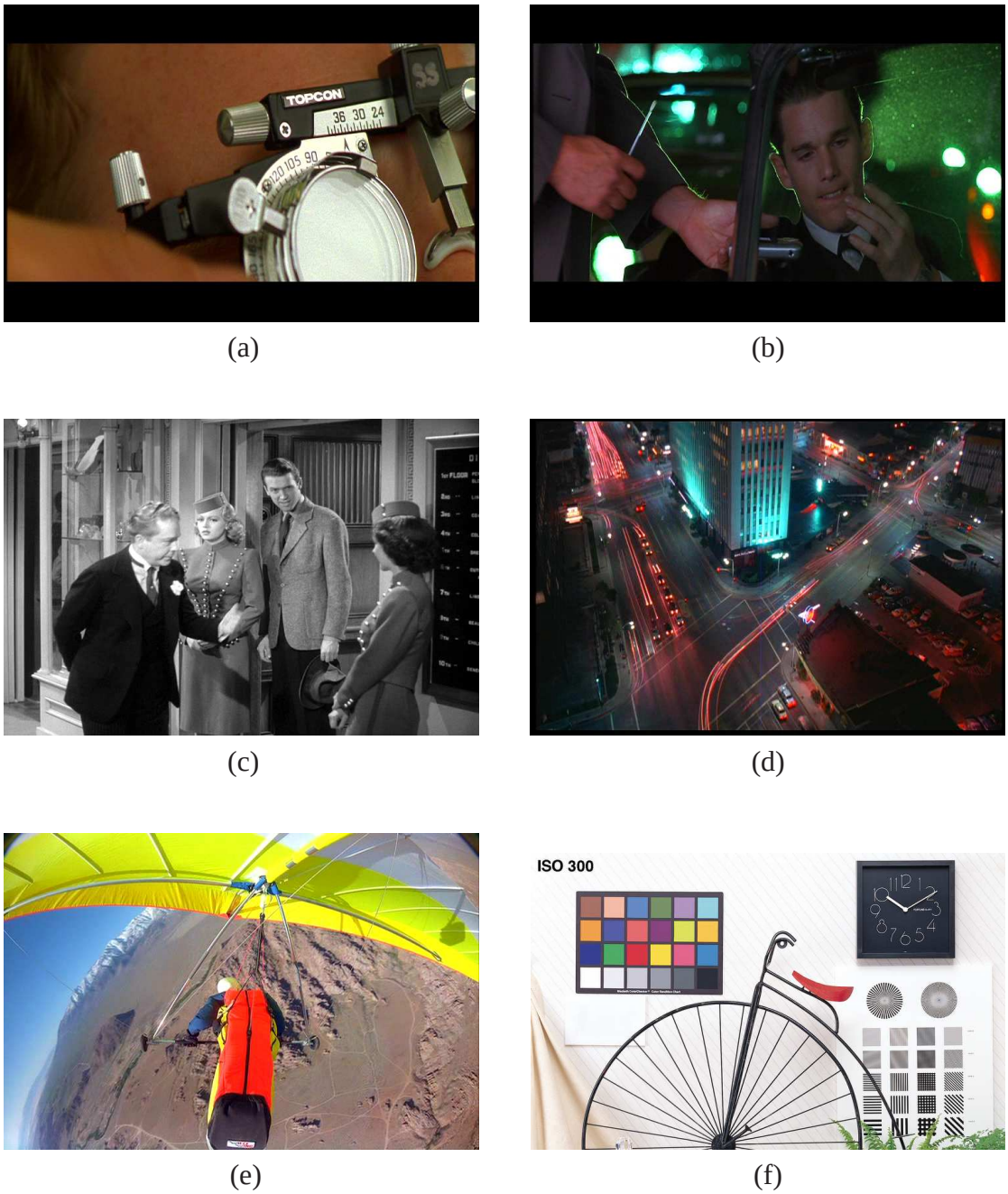
On the algorithmic side, all training samples are labeled and grouped with respect to their context classes *and* phase numbers. To decide on the number of phases to be used for a specific scaling ratio, we experiment with an increasing number of phases until we obtain visually satisfactory results. For the scaling ratios typically required in SD to HD conversion ($1.5 < L_x, L_y < 2.5$), we observed that at most six phases provided visually satisfactory results.

4.4 Experimental setup and simulation results

In this section we present visual results to demonstrate the performance of the modified resolution synthesis algorithm. Before we discuss the details of the experimental setup, we note that for all of the experiments training and test data were kept completely separate, *i.e.*, none of the images or video frames used in off-line training was used for testing purposes. The test images are shown in Figures 4.18 and 4.19; and the training images are shown in Figure 4.20. Images shown in Figures 4.18(a), (b), (c), (d), (e), 4.20(c) and (d) are 720×480 frames captured from DVD movies. The images shown in Figures 4.18(f), 4.19(a), (b), (c), (d), 4.20(a) and (b) are still images of natural scenes.[10]

In the single-frame resolution enhancement case visual performance is defined by the quality and sharpness of edges, *i.e.*, smooth along the edge without jaggies with a sharp and ringing free transition across the edge. Unfortunately, such visual attributes can not be assessed accurately by the currently available numeric metrics, hence we shall not present any numeric results. For visual comparisons we have results obtained by three methods,

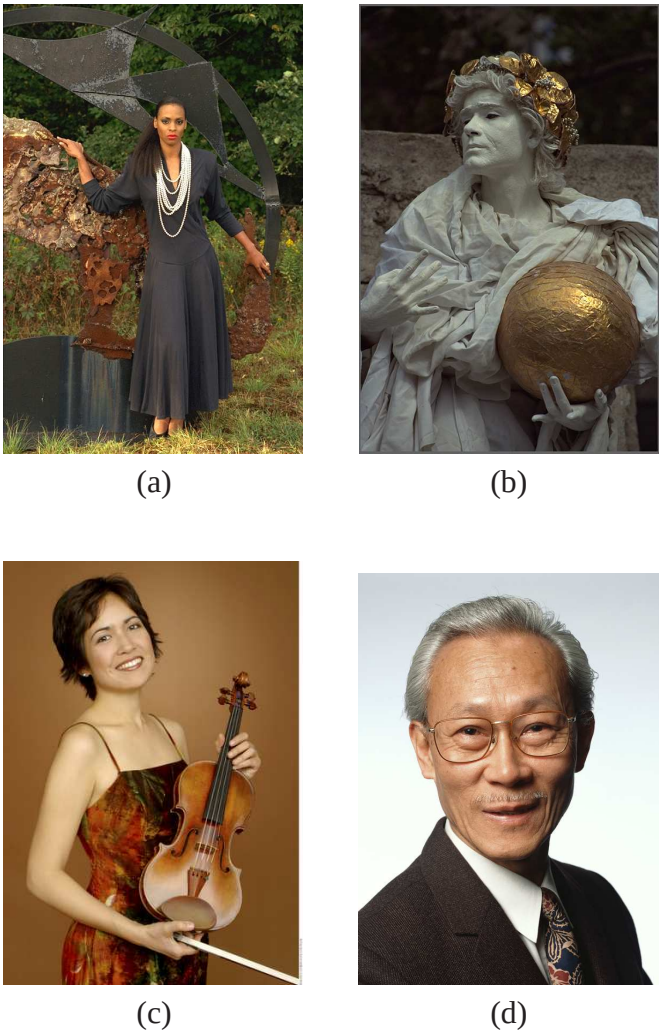
[10]Note that these images are downscaled to allow a clear presentation.



(a) (b) (c) (d) (e) (f)

Figure 4.18. Test pictures 1 through 6

namely, bicubic interpolation, original resolution synthesis (ORS) with 11 context classes, and modified resolution synthesis (MRS) with 11 context classes. The ORS and MRS methods were trained on the exact same training set (approximately 500,000 registered high-low resolution pixels) extracted from the images shown in Figure 4.20. The ORS uses soft-filtering, *i.e*, output pixels are computed as weighted combinations of eleven context



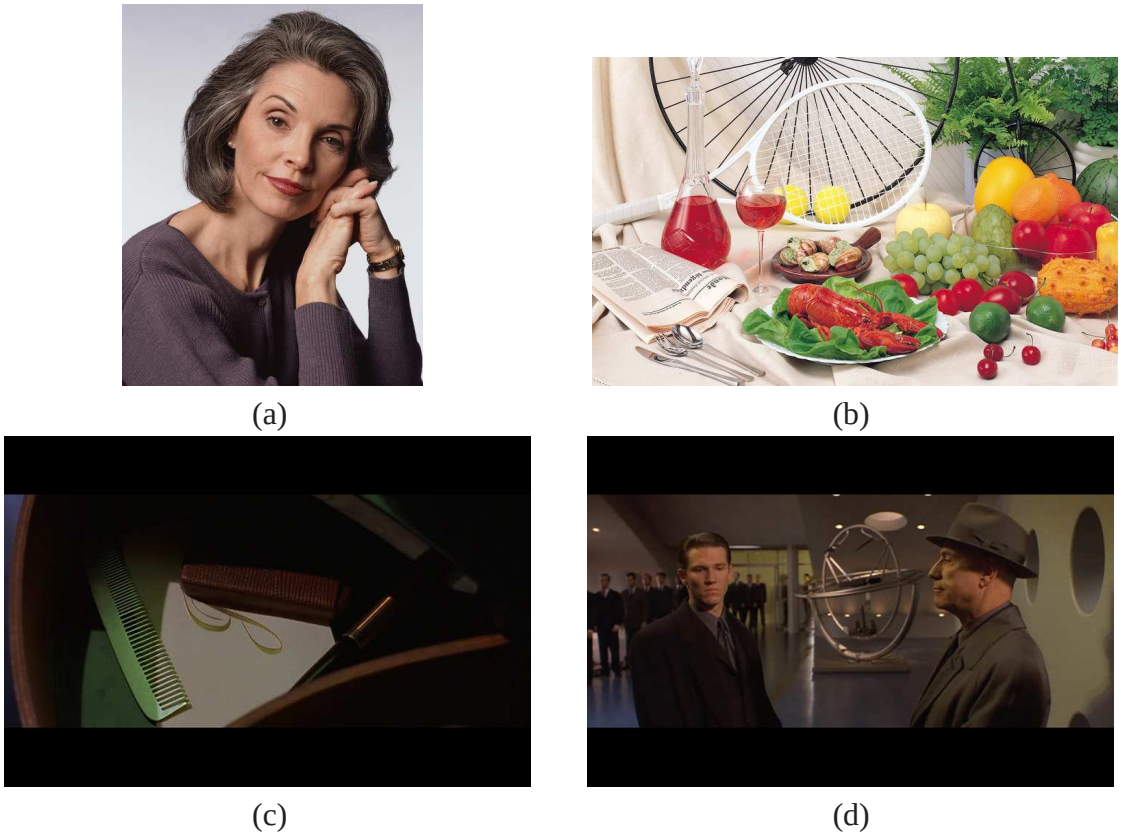
(a) (b)

(c) (d)

Figure 4.19. Test pictures 7 through 10

classes. Each ORS class output is computed by applying a 5×5 filter mask and adding a bias term as described in [51]. The MRS method uses hard-decision, *i.e.*, the filter of the best matching context class. The MRS method uses 4×4 filter masks without any bias terms. Figures 4.21 and 4.22 show a comparison of ORS and MRS. From the visual results we can see that MRS can achieve satisfactory visual performance with as low as eleven context classes using 4×4 filters with hard-decision.

Figures 4.23 and 4.24 show the results of upscaling with $L = 2$ and $L = 2.5$, respectively. The improvement in edge definition and elimination of the jaggy patterns are



(a) (b)

(c) (d)

Figure 4.20. Training pictures

obvious. From the visual results, we can see that MRS can outperform bicubic interpolation with as low as eleven context classes using 4×4 filters with hard-decision. The result presented in Figure 4.24 (corresponding to $L = 2.5$) also shows that the proposed generalization method for handling non-integer scaling ratios achieves satisfactory visual performance.

For subjective quality evaluation we conducted two experiments. Five test images were upscaled by two, and specific regions where the visual improvements are clearly visible are presented. The test video clip is captured from the DVD movie Gattaca. We did not postprocess the captured clip (denoising, sharpening and additional scaling) apart from clipping the top and bottom letterboxes. The video clip is upscaled by two from an initial resolution of 640×320 to 1280×640. The upscaled images and the video clip were presented to a group of 10 viewers and the viewers were asked to order the images in terms

(a) (b)

Figure 4.21. (a) Original resolution synthesis and (b) modified resolution synthesis.

(a) (b)

Figure 4.22. (a) Original resolution synthesis and (b) modified resolution synthesis.

(a)

(b)

(c)

Figure 4.23. Scaling results for $L = 2$. (a)Original (b)Bicubic (c)MRS

of visual quality. All viewers were fairly knowledgable in the field of digital image/video processing. In all cases, we have the result of MSR, ORS with soft-filtering, and bicubic

interpolation. All viewers reported that MRS and original RS with soft-filtering are visually more preferable over bicubic interpolation and RS with hard-filtering. The visual quality difference between MRS and RS with soft-filtering was not clearly distinguishable to most of the viewers.

(a)

(b)

(c)

Figure 4.24. Scaling results for $L = 2.5$. (a)Original (b)Bicubic (c)MRS

CHAPTER 5

RESOLUTION ENHANCEMENT OF HYPERSPECTRAL IMAGERY

The goal of our research is to enhance spatial resolution of hyperspectral images. An integral part of our approach is a model of the hyperspectral image acquisition process. We require our model to be complex enough to capture the main characteristics of the imaging process, while keeping it as simple as possible to keep its computational complexity within practical limits. Although the proposed model makes no specific assumptions about the imaging device used and incorporates most of the effects that influence the spatial and spectral resolution of the observed scene, it excludes mainly the physical effects. These are related to the sensor characteristics and secondary illumination sources. In our work, we assume that sensor calibration and atmospheric compensation have already been applied, and focus on the image processing aspects of the acquisition.

Our hyperspectral image acquisition model interprets source images (also referred as observations) as aliased and optically blurred linear combinations of the target image's[1] basis image planes. The pixel values of these basis image planes correspond to the principle component magnitudes. This section provides a detailed mathematical formulation of the proposed model. In the next section, we will address the inverse problem and present a back-projections-based iterative solution method. Possible simplifications for single observation and multiple observations with translational motion will be studied and a useful interpretation of the overall imaging process will be presented.

For a given ground pixel, whose dimensions can be in the range of tens of centimeters to tens of meters depending on the spatial resolution and altitude of the imaging device,

[1]Note that we use the term *target image* to denote the high resolution image cube which we are trying to reconstruct. In a similar fashion, the term *source image* denotes the low resolution observation which is available to us.

the radiance observed at any particular wavelength is determined, to first order, by the reflectance of the matter and the solar illumination at that wavelength. But there are many important secondary effects that limit the measurement, including scattering and absorption of the reflected radiance by the atmosphere, spatial and spectral aberrations in the sensors, imperfect optics in the imaging device, secondary illumination from adjacent objects, finite sensor dimensions, and the viewing angle of the sensor array. Characterizing these effects with the ultimate goal of developing compensation techniques to limit their undesired influence on the image data is a challenging problem and an active research area. There are many different models that describe the hyperspectral images. Statistical models [56], [57], [58] typically use some kind of Markov random field (for example, Gauss-Markov random fields) and are capable of capturing the spatially and spectrally correlated nature of hyperspectral data. Deterministic models on the other hand are computationally more attractive and can easily be structured to have guidance of a physical model of the imaging process [59], [60]. The deterministic models can be further divided into two subgroups, namely, linear deterministic models and non-linear deterministic models [61]. Finally, there are approaches that combine the statistical and deterministic models in an effort to construct models that have the advantages of both approaches [62].

5.1 The hyperspectral image acquisition model

In following sections we model the image acquisition, spatial filtering, spectral filtering, and sampling. We begin with a summary of the mathematical notation that will be used throughout the remaining of this thesis. The hyperspectral image data is best represented as an R-dimensional vector for each pixel, where R is the number of spectral bands. The images are assumed to be $N_1 \times N_2$ so that the hyperspectral data forms an $N_1 \times N_2 \times R$ data cube. Following this convention we let $\boldsymbol{f}[\boldsymbol{n}] = [f_1[\boldsymbol{n}]\ f_2[\boldsymbol{n}]\ \ldots\ f_R[\boldsymbol{n}]]^T$ denote the R-dimensional pixel value at location $\boldsymbol{n} = [n_1, n_2]^T$. We use $f_j(x_1, x_2)$ to denote the j^{th} spatially continuous high resolution (*target*) image plane and $f_j[n_1, n_2]$ for the j^{th} spatially discrete

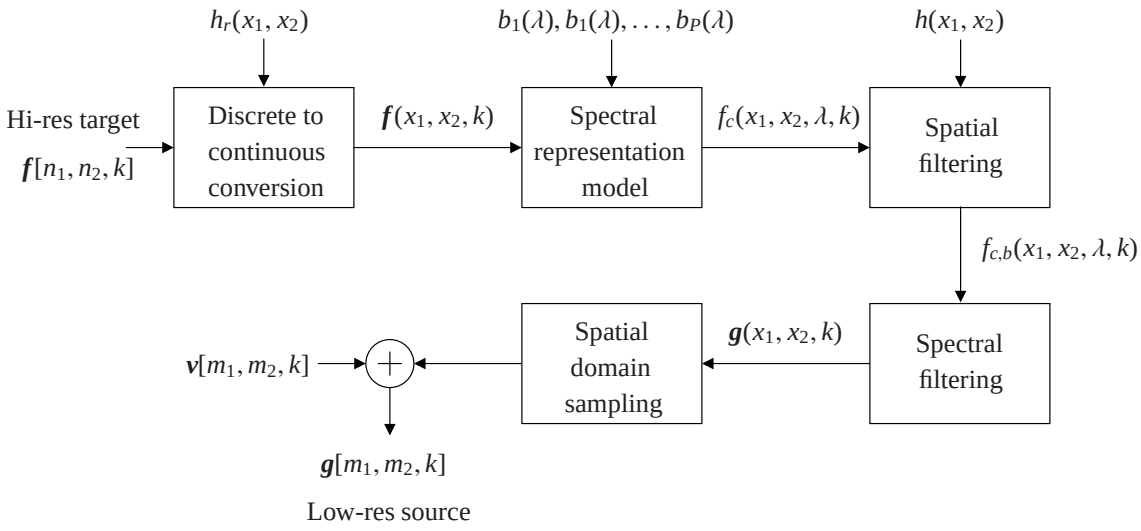


Figure 5.1. The hyperspectral image acquisition model by which a hypothesized high resolution discrete target hyperspectral image is used to produce a low resolution source hyperspectral image

high resolution image plane. Similarly $g_i(x_1, x_2)$ denotes the i^{th} continuous low resolution (*source*) image plane and $g_i[m_1, m_2]$ denotes the i^{th} discrete low resolution image plane. Any pixel denoted by the letter f, no matter what its subscript or indices may be, is a target image pixel. The letter g similarly always denotes a source image pixel. Furthermore, at some point it will be necessary to differentiate between high and low resolution grid pixels. For this purpose, the high resolution grid pixels are indexed with $\boldsymbol{n} = [n_1, n_2]^T$ and the low resolution grid pixels are indexed with $\boldsymbol{m} = [m_1, m_2]^T$. A complete list of terms and their definitions is given in Table 5.1.

The block diagram shown in Figure 5.1 depicts the system to be modeled. The ideal continuous-space and continuous-spectrum image signal, denoted by $f_c(x_1, x_2, \lambda, k)$, represents the actual input to the imaging device. In this notation k is the observation index. Our main assumption in superresolution reconstruction is that we have access to multiple observations of the scene for which we wish to apply superresolution. These observations can be hyperspectral images captured at different times[2] by a single imaging device or simultaneously from multiple imaging devices. Superresolution reconstruction then fuses

[2]Please note that this is not an implicit assumption of the existence of a fictitious *hyperspectral video signal*. The observations can be captured at time instances which are separated by arbitrarily long time periods.

the information present across these observations to obtain a higher resolution image of the target scene [11], [63]. Ideally, we would like to reconstruct $f_c(x_1, x_2, \lambda, k)$ from the available observations, but $f_c(x_1, x_2, \lambda, k)$ is continuous in all dimensions and there is no way we can implement a solution to this problem using digital hardware. We will deal with this limitation in two steps. First, we will consider the spectral dimension, where we will make use of a well known and widely used property of hyperspectral image data. Then, we will look into the spatial dimension.

Table 5.1. List of terms

Term	Definition
N_1	horizontal dimension of the high resolution target
N_2	vertical dimension of the high resolution target
M_1	horizontal dimension of the low resolution source
M_2	vertical dimension of the low resolution source
R	number of the spectral bands originally present in the target images
Q	number of the spectral bands present in the source
P	number of the spectral basis functions
$\boldsymbol{x} = [x_1, x_2]^T$	continuous spatial index
$\boldsymbol{n} = [n_1, n_2]^T$	high resolution grid index
$\boldsymbol{m} = [m_1, m_2]^T$	low resolution grid index
$h_r(x_1, x_2)$	reconstruction filter
$h(x_1, x_2)$	spatial blur filter
$h_b(\boldsymbol{x}; \boldsymbol{n}; k; k_r)$	generalized blur filter
$f_j(x_1, x_2, k)$	j^{th} continuous target plane of the k^{th} observation
$f_j[n_1, n_2, k]$	j^{th} discrete target plane of the k^{th} observation
$g_i(x_1, x_2, k)$	i^{th} continuous source plane of the k^{th} observation
$g_i[m_1, m_2, k]$	i^{th} discrete source plane of the k^{th} observation
$p_j(\lambda)$	illuminant-independent spectral basis functions
$b_j(\lambda)$	illuminant-dependent spectral basis functions
L_1	downsampling ratio in the horizontal direction
L_2	downsampling ratio in the vertical direction
$v_i[m_1, m_2]$	noise process
$\boldsymbol{W}$	basis weighting matrix
$\boldsymbol{H}$	spatial blur filter
$\boldsymbol{B}$	combined blur and weight matix
$\boldsymbol{B}_{i,j}$	stands for the element of $\boldsymbol{B}$ located at the i^{th} row and the j^{th} column

5.1.1 Discretizing the target image

It is a well known fact that the spectral reflectance of natural images can be accurately modelled using linear combinations of a relatively small number (generally around seven, [64])

of reflectance basis functions, $p_1(\lambda), \ldots, p_P(\lambda)$. These illuminant-independent orthonormal basis functions can be obtained by applying PCA (Principal Components Analysis) to a large set of natural image reflectances and selecting the first P principal components. If we denote the illuminant spectrum as $L(\lambda)$, then one possible choice for a set of illuminant-dependent basis functions is $b_i(\lambda) = L(\lambda)p_i(\lambda)$. As a first step in our model we will assume that $f_c(x_1, x_2, \lambda, k)$ is representable as a linear combination of these basis functions. That is, at every location, $f_c(x_1, x_2, \lambda, k)$ will be represented by a P-dimensional vector, where the elements of this vector are the coefficients of the corresponding orthonormal basis functions. Therefore, the high resolution target image, $\boldsymbol{f}[n_1, n_2]$, shown in Figure 5.1 is a P-dimensional vector at every pixel. Note that P is not the number of spectral bands; it is the number of spectral basis functions. This assumption lets us represent an R-dimensional signal in a P-dimensional space (note that $R \gg P$) with negligible error. This greatly reduces the complexity of the reconstruction problem.

Before starting to discuss the spatial domain, we would like to comment on the use of PCA in the context of resolution improvement and the spectral information fusion aspect of the proposed technique. First of all, our main assumption in attempting to fuse information coming from multiple spectral bands is that the spectral signature of some target material we are interested in is present in several bands. No claims are made for spectral details that may be present only in a single frequency band of a single observation. Second, the choice of basis functions is application specific. If we are trying to improve the resolution of a specific material with a known spectral signature, then the training images can be chosen accordingly to have basis vectors optimized for that specific material. Also at the expense of increased computational load, the number of the basis functions used to represent $f_c(x_1, x_2, \lambda, k)$ can be increased and the representation error can be made arbitrarily small. Finally, the use of PCA to find the spectral basis functions is totally arbitrary. In fact, the basis functions may be calculated using a variety of approaches including but not limited to, convex geometry-based approaches, noise reduction-based approaches, etc.

(see [65] for a detailed discussion of the available techniques).

To deal with the spatial domain, we hypothesize that for each of the P spectral basis image planes, there exists a corresponding discrete, high-resolution target image plane $f_j[n_1, n_2, k]$ ($j = 1, 2, \ldots, P$) and we seek to reconstruct a target image from that signal, $f_j(n_1, n_2, k_r)$. The main assumption here is that the spatially continuous signal $f_j(x_1, x_2, k)$ is bandlimited (more details on the band-limitedness assumption will be given in the next section) and therefore could be reconstructed from the spatially discrete high-resolution image $f_j[n_1, n_2, k]$ through an ideal reconstruction filter h_r.

5.1.2 Discrete-to-continuous conversion

The first step in the ideal reconstruction process is conversion of the discrete signals into impulse trains. The following operations are performed on each of the P target image planes. If we let $f_{s,j}(x_1, x_2, k)$ denote the impulse array obtained from $f_j[n_1, n_2, k]$, then we can write

$$f_{s,j}(x_1, x_2, k) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k]\delta(x_1 - \frac{n_1}{L_1}, x_2 - \frac{n_2}{L_2}). \tag{5.1}$$

Note that the spatial sampling frequency is normalized for the low resolution grid so that L_1 and L_2 show the increase in the spatial sampling density when we move from the low resolution image (source) to the high resolution image (target). In other words, if we assume that the sampling density in the low resolution image is 1 per unit area, then the high resolution image has L_1 and L_2 samples in the horizontal and vertical directions per unit area, respectively. Under this normalization our band-limitedness assumption requires the continuous signal $f_j(x_1, x_2, k)$ to be bandlimited to the frequency range $(-L_1\pi, L_1\pi) \times (-L_2\pi, L_2\pi)$.

We implicity assume that the high resolution target image (and hence its reconstructed version) exists for all observations k. Therefore, in the following equations the observation index k is suppressed. Keeping this in mind, the convolution with the reconstruction filter takes the familiar form

$$f_j(x_1, x_2) = \iint f_{s,j}(x_1 - u_1, x_2 - u_2)h_r(u_1, u_2)\, du_1 du_2. \tag{5.2}$$

Substituting for $f_{s,j}(x_1, x_2)$ from Eq. 5.1 we get

$$f_j(x_1, x_2) = \iint \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2] \delta(x_1 - u_1 - \frac{n_1}{L_1}, x_2 - u_2 - \frac{n_2}{L_2}) h_r(u_1, u_2)\, du_1 du_2. \quad (5.3)$$

Assuming convergence, we can exchange the order of summation and integration to write

$$f_j(x_1, x_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2] \iint \delta(x_1 - u_1 - \frac{n_1}{L_1}, x_2 - u_2 - \frac{n_2}{L_2}) h_r(u_1, u_2)\, du_1 du_2 \quad (5.4)$$

$$= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2] h_r(x_1 - \frac{n_1}{L_1}, x_2 - \frac{n_2}{L_2}). \quad (5.5)$$

If we include the suppressed observation index k, Eq. 5.4 becomes

$$f_j(x_1, x_2, k) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k] h_r(x_1 - \frac{n_1}{L_1}, x_2 - \frac{n_2}{L_2}). \quad (5.6)$$

5.1.3 Spectral representation with predetermined basis functions

We assume that the basis functions have been predetermined by applying PCA on appropriate training data and selecting the first P principal components. As mentioned in Section 5.1.1, the use of PCA is arbitrary and any of the methods mentioned in [65] can be used to obtain these basis functions. If we denote the continuous signal as $f_c(x_1, x_2, \lambda, k)$ then we have

$$f_c(x_1, x_2, \lambda, k) = \sum_{j=1}^{P} b_j(\lambda) f_j(x_1, x_2, k). \quad (5.7)$$

Noting that Eq. 5.6 applies to each of the P target image planes, we can write

$$f_c(x_1, x_2, \lambda, k) = \sum_{j=1}^{P} b_j(\lambda) \left[\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k] h_r(x_1 - \frac{n_1}{L_1}, x_2 - \frac{n_2}{L_2}) \right]. \quad (5.8)$$

Before we move on, we would like to point out a connection between this work and a previous work on face superresolution by Gunturk *et al.* [33]. There are some major differences between these works, such as the fact that the previous work used spatial basis whereas here the basis are spectral. Nevertheless, the undeniable similarities call for a comparison. In both methods, the use of the low-dimensional space to which the unknown high resolution images are known to belong, serve as an effective regularization.

Furthermore, both methods take the advantage of projecting the noise process to a lower dimensional space which in turn reduces its undesired effects on the reconstructed high resolution images.

5.1.4 Spatial filtering

We use $h(x_1, x_2)$ to denote the spatially invariant blur filter. This models the imperfect imaging optics (e.g. lens blur) and the unavoidable sensor integration blur caused by the finite sensor area. In the following derivation we assume that the blur filters for all the spectral basis functions are the same. This is justified by the spatial response functions supported with the AVIRIS data and will lead to a relatively simple final relation between the high resolution target image and the low resolution observations. Please note that the solution method that will be used to obtain the target image can handle different blur filters for every basis function with only minor modifications (more on this in Section 5.2). The blur operation can be written as the convolution of the target image planes with the point spread function of blur filter

$$f_{c,b}(x_1, x_2, \lambda, k) = \iint h(x_1 - \nu_1, x_2 - \nu_2)\, f_c(\nu_1, \nu_2, k, \lambda) d\nu_1 d\nu_2, \tag{5.9}$$

where subscript c, b means *continuous and blurred*. We will use the motion mapping $\boldsymbol{M}$ for relating the available observations to the reference observation [42]. $\boldsymbol{M} = (M_1, M_2)$ is defined as

$$\begin{aligned} x_{1,r} &= M_1(x_1, x_2, k, k_r), \\ x_{2,r} &= M_2(x_1, x_2, k, k_r). \end{aligned} \tag{5.10}$$

Let us assume that the pixel located at $(x_{1,r}, x_{2,r})$ in observation k_r corresponds to (κ_1, κ_2) in observation k. That is, $f_c(x_{1,r}, x_{2,r}, k_r, \lambda) = f_c(\kappa_1, \kappa_2, k, \lambda)$. Then by using the inverse of the mapping mentioned above, we can write $f_{c,b}(x_1, x_2, \lambda, k)$ in terms of $f_c(x_{1,r}, x_{2,r}, k_r, \lambda)$.

$$\begin{aligned} f_{c,b}(x_1, x_2, \lambda, k) = \iint h(x_1 - M_1^{-1}(x_{1,r}, x_{2,r}, k, k_r), x_2 - M_2^{-1}(x_{1,r}, x_{2,r}, k, k_r))|\boldsymbol{J}| \\ \times f_c(x_{1,r}, x_{2,r}, k_r, \lambda) dx_{1,r} dx_{2,r} \end{aligned} \tag{5.11}$$

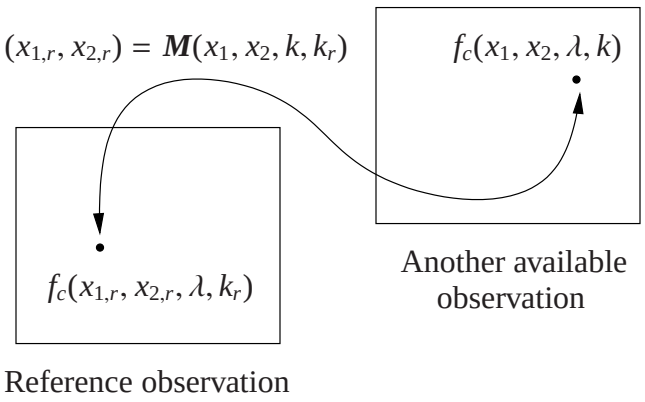


Figure 5.2. The motion mapping M relates the available observations to the reference observation.

where $|\boldsymbol{J}|$ is the Jacobian of the motion mapping. To leave no room for misunderstanding, we note that the inverse mapping $\boldsymbol{M}^{-1} = (M_1^{-1}, M_2^{-1})$ maps a given pixel in observation k_r back to its location in observation k, that is

$$\begin{aligned} x_1 &= M_1^{-1}(x_{1,r}, x_{2,r}, k, k_r), \\ x_2 &= M_2^{-1}(x_{1,r}, x_{2,r}, k, k_r). \end{aligned} \tag{5.12}$$

If we define h_M as

$$h_M(x_1, x_2; x_{1,r}, x_{2,r}; k; k_r) \triangleq |\boldsymbol{J}| h(x_1 - M_1^{-1}(x_{1,r}, x_{2,r}, k, k_r), x_2 - M_2^{-1}(x_{1,r}, x_{2,r}, k, k_r)), \tag{5.13}$$

$f_{c,b}(x_1, x_2, \lambda, k)$ can be written as follows:

$$f_{c,b}(x_1, x_2, \lambda, k) = \iint h_M(x_1, x_2; x_{1,r}, x_{2,r}; k; k_r) f_c(x_{1,r}, x_{2,r}, k_r, \lambda) dx_{1,r} dx_{2,r}. \tag{5.14}$$

Substituting from Eq. 5.8 for $f_c(x_1, x_2, \lambda, k)$ into this expression we get

$$\begin{aligned} f_{c,b}(x_1, x_2, \lambda, k) = \iint & h_M(x_1, x_2; x_{1,r}, x_{2,r}; k; k_r) \\ & \times \left[\sum_{j=1}^{P} b_j(\lambda) \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] h_r(x_{1,r} - \frac{n_1}{L_1}, x_{2,r} - \frac{n_2}{L_2}) \right] dx_{1,r} dx_{2,r}. \end{aligned} \tag{5.15}$$

Again, assuming convergence we can exchange the integration and summations to obtain

$$f_{c,b}(x_1, x_2, \lambda, k) = \sum_{j=1}^{P} b_j(\lambda) \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] \times \iint h_M(x_1, x_2; x_{1,r}, x_{2,r}; k; k_r) h_r(x_{1,r} - \frac{n_1}{L_1}, x_{2,r} - \frac{n_2}{L_2})\, dx_{1,r} dx_{2,r}. \tag{5.16}$$

To get a simpler looking expression for $f_{c,b}(x_1, x_2, \lambda, k)$ we define h_b as

$$h_b(x_1, x_2; n_1, n_2; k; k_r) \triangleq \iint h_M(x_1, x_2; x_{1,r}, x_{2,r}; k; k_r) h_r(x_{1,r} - \frac{n_1}{L_1}, x_{2,r} - \frac{n_2}{L_2}) dx_{1,r} dx_{2,r}, \tag{5.17}$$

which allows us to write

$$f_{c,b}(x_1, x_2, \lambda, k) = \sum_{j=1}^{P} b_j(\lambda) \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] h_b(x_1, x_2; n_1, n_2; k; k_r). \tag{5.18}$$

5.1.5 Spectral filtering: band selection, atmospheric and illuminator based effects on spectrum)

The spectral response functions, $r_1(\lambda),\ r_2(\lambda),\ \ldots,\ r_Q(\lambda)$, where Q stands for the number of spectral bands in the source images, model the hyperspectral sensors' efficiency at different wavelengths. We assume that the input images are atmospherically corrected, which in turn eliminates the need for complex processing to invert the atmospheric effects on the spectrum.

$$\begin{aligned} g_i(x_1, x_2, k) &= \int_0^{\infty} f_{c,b}(x_1, x_2, \lambda, k) r_i(\lambda)\, d\lambda \\ &= \int_0^{\infty} \left[\sum_{j=1}^{P} b_j(\lambda) \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] h_b(x_1, x_2; n_1, n_2; k; k_r) \right] r_i(\lambda)\, d\lambda \\ &= \sum_{j=1}^{P} \left[\int_0^{\infty} b_j(\lambda) r_i(\lambda)\, d\lambda \right] \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] h_b(x_1, x_2; n_1, n_2; k; k_r), \end{aligned} \tag{5.19}$$

where the second equality follows from the assumption that the integrals and summations converge, allowing us to change their order. If we denote the integral in brackets as $w_{i,j}$, then we can write

$$g_i(x_1, x_2, k) = \sum_{j=1}^{P} w_{i,j} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] h_b(x_1, x_2; n_1, n_2; k; k_r) \quad \text{for } i = 1, \ldots, Q. \tag{5.20}$$

From Eq. 5.20 we see that, within the limitations of our model, the low resolution source images can be represented as linear combinations of the basis planes filtered by h_b. The weights are obtained by separately applying the spectral filters to the basis functions. Figure 5.3 demonstrates this for the weights of the i^{th} source plane with a fictitious spectrum.

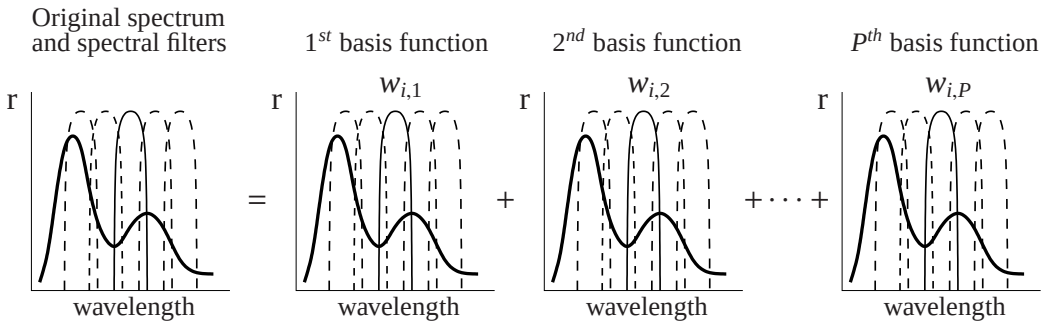


Figure 5.3. Spectral filtering: The i^{th} spectral filter (solid line) is applied to all basis functions to produce the weights of the j^{th} source plane, $w_{i,j}$.

5.1.6 Spatial domain sampling

Next we must spatially discretize the images to make a practical implementation possible. This is done by sampling the g_i's on a low resolution $M_1 \times M_2$ grid.

$$\begin{aligned} g_i[m_1, m_2, k] &= g_i(x_1, x_2, k)\Big|_{x_1=m_1,\ x_2=m_2} \\ &= \sum_{j=1}^{P} w_{i,j} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] h_b[m_1, m_2; n_1, n_2; k; k_r] \end{aligned} \tag{5.21}$$

or in matrix form with $Q = 112$ and $P = 6$

$$\begin{bmatrix} g_1[\boldsymbol{m}, k] \\ g_2[\boldsymbol{m}, k] \\ \vdots \\ g_{112}[\boldsymbol{m}, k] \end{bmatrix} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,6} \\ w_{2,1} & \cdots & w_{2,6} \\ \vdots & \ddots & \vdots \\ w_{112,1} & \dots & w_{112,6} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1[\boldsymbol{n}, k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m}; \boldsymbol{n}; k; k_r] \\ \boldsymbol{f}_2[\boldsymbol{n}, k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m}; \boldsymbol{n}; k; k_r] \\ \vdots \\ \boldsymbol{f}_6[\boldsymbol{n}, k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m}; \boldsymbol{n}; k; k_r] \end{bmatrix}$$

$$\boldsymbol{g} = \boldsymbol{W}\boldsymbol{H}\boldsymbol{f}, \tag{5.22}$$

where we have made the following definitions to simplify the expression:

$$\boldsymbol{f}_j[\boldsymbol{n}, k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m}; \boldsymbol{n}; k; k_r] \triangleq \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] h_b[m_1, m_2; n_1, n_2; k; k_r], \tag{5.23}$$

$$\boldsymbol{f}[\boldsymbol{n}, k_r] \triangleq \begin{bmatrix} \boldsymbol{f}_1[\boldsymbol{n}, k_r] \\ \boldsymbol{f}_2[\boldsymbol{n}, k_r] \\ \vdots \\ \boldsymbol{f}_6[\boldsymbol{n}, k_r] \end{bmatrix}, \quad \boldsymbol{H}\boldsymbol{f}[\boldsymbol{m}, k] \triangleq \begin{bmatrix} \boldsymbol{f}_1[\boldsymbol{n}, k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m}; \boldsymbol{n}; k; k_r] \\ \boldsymbol{f}_2[\boldsymbol{n}, k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m}; \boldsymbol{n}; k; k_r] \\ \vdots \\ \boldsymbol{f}_6[\boldsymbol{n}, k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m}; \boldsymbol{n}; k; k_r] \end{bmatrix},$$

$$\boldsymbol{g}[\boldsymbol{m}, k] \triangleq \begin{bmatrix} g_1[\boldsymbol{m}, k] \\ g_2[\boldsymbol{m}, k] \\ \vdots \\ g_{112}[\boldsymbol{m}, k] \end{bmatrix}, \quad \boldsymbol{W} \triangleq \begin{bmatrix} w_{1,1} & \cdots & w_{1,6} \\ w_{2,1} & \cdots & w_{2,6} \\ \vdots & \ddots & \vdots \\ w_{112,1} & \ldots & w_{112,6} \end{bmatrix}, \text{ for } Q = 112 \text{ and } P = 6.$$

Note that $\boldsymbol{f}_j[\boldsymbol{n}, k_r]$ (for $j = 1, \ldots, P$) and $\boldsymbol{h}_b[\boldsymbol{m}; \boldsymbol{n}; k; k_r]$ as defined above, are vectors obtained by cascading the corresponding elements in definition 5.23 one after another. Eq. 5.21 shows the relationship between the low resolution observations and the high resolution target image cube through the discrete (spatially) shift-varying blur function $h_b[m_1, m_2; n_1, n_2; k; k_r]$.

5.1.7 Additive noise

Finally, the additive noise, $v[m_1, m_2, k]$, models the total effect of all possible noise sources (unavoidable sensor noise, sampling noise, quantization noise introduced when the sampled pixel values are quantized) that exist throughout the whole acquisition process,

$$g_i[m_1, m_2, k] + v_i[m_1, m_2, k] \quad \text{for } i = 1, \ldots, Q. \tag{5.24}$$

The exact statistical nature of the noise process, which is of great importance for methods formulated using a Bayesian framework, depends on the specific application and the assumptions we are willing to make. A very popular characterization is to assume that $\boldsymbol{v}[m_1, m_2]$ is a zero mean wide sense stationary Gaussian noise process.

5.2 The inverse problem

Given the model presented in the previous sections, the inverse problem can be stated as finding the target image that agrees with the available source images. Here *agrees* deserves some explanation. When we say the candidate target image is in agreement with the source images we mean that if the linear, time and space-varying (LTSV) filter h_b in Eq. 5.20 is applied to the candidate target image, the resulting synthetic source image is close to the actual images captured by the imaging device under consideration. There exist many ways to solve this problem, each with its (dis)advantages. For example, we could try to minimize the squared error between the observed images and the synthetically produced source images by using well-studied least squares methods. The drawback of this approach is that it requires the computation of the inverses of large matrices, which is in most cases very difficult.

A preferable alternative is to use iterative set-theoretic methods, [42], [66]. It can be shown that using a squared-error criterion together with a gradient based iterative minimization method is completely equivalent to a version of the Back-Projection method [67]. In this work, we propose a POCS (Projection onto Convex Sets, see [68], [69]) based solution to the inverse problem addressed above. The POCS method requires a number of closed convex constraint sets to be defined. These constraint sets must be defined in a well-defined vector space and contain the high-resolution (hi-res) target image. We define Q constraint sets (one for each observed band) at every low-res grid point $[m_1, m_2, k]$ where Eq. 5.21 is valid. A reconstructed hi-res image is a point in the intersection of these constraint sets and can be determined by successively projecting an initial estimate (which is usually chosen to be a bilinearly interpolated low-res image) onto the constraint sets. As mentioned in almost every work that applies a POCS based reconstruction method, we require an accurate estimate of the motion field for this approach to work. Otherwise, the projection operators dictate irrelevant constraints on the pixels with inaccurate motion vectors, which results in a degradation in the image quality. Fortunately, for hyperspectral

images complex motion fields with high motion rates and frequent occlusion regions are quite rare due to the nature of the data.

As in [42] and [66], we start with defining the following closed, convex constraint sets, for each low-res grid pixel:

$$C_i(m_1, m_2, k) = \{\tilde{f}_j[n_1, n_2, k_r] \text{ for } j = 1, \ldots, P : |d_i^k[m_1, m_2]| < \delta_i^k[m_1, m_2]\}, \text{ for } i = 1, \ldots, Q, \tag{5.25}$$

where

$$d_i^k[m_1, m_2] \triangleq g_i[m_1, m_2, k] - \sum_{j=1}^{P} w_{i,j} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \tilde{f}_j[n_1, n_2, k_r] h_b(m_1, m_2; n_1, n_2; k; k_r), \tag{5.26}$$

is the residual signal associated with the k^{th} observation of the i^{th} spectral band. The quantities $\delta_i^k[m_1, m_2]$ used in the definition of the constraint set $C_i(m_1, m_2, k)$ reflect our statistical confidence with which the actual hi-res target $f_j[n_1, n_2, k_r]$ (for $j = 1, \ldots, P$) is in $C_i(m_1, m_2, k)$, see [66]. We can determine the values of $\delta_i^k[m_1, m_2]$'s from the statistics of the noise process to guarantee that the actual hi-res target is an element of the constraint set with some pre-determined statistical confidence.

To determine the projection operator $P_{C_i(m_1,m_2,k)}$ that projects the current estimate of the hi-res target $\tilde{f}_j[n_1, n_2, k_r]$ (for $j = 1, \ldots, P$) onto $C_i(m_1, m_2, k)$, we start with Eq. 5.22. Combining $\boldsymbol{W}$ and $\boldsymbol{H}$ into a single matrix $\boldsymbol{B}$, we obtain:

$$\boldsymbol{g} = \boldsymbol{WHf} = \boldsymbol{Bf}. \tag{5.27}$$

The projection operator is then defined as:

$$P_{C_i(m_1,m_2,k)}[\tilde{f}_j[n_1, n_2, k_r]] = \tilde{f}_j[n_1, n_2, k_r] + \begin{cases} \gamma(d_i^k[m_1, m_2] - \delta_i^k[m_1, m_2]), & \text{if } d_i^k[m_1, m_2] > \delta_i^k[m_1, m_2]. \\ \gamma(d_i^k[m_1, m_2] + \delta_i^k[m_1, m_2]), & \text{if } d_i^k[m_1, m_2] < -\delta_i^k[m_1, m_2]. \\ 0, & \text{otherwise.} \end{cases}$$

for $j = 1, \ldots, P$, where we used

$$\gamma \triangleq \frac{\boldsymbol{B}_{i,j}}{\sqrt{\sum_{j=1}^{P} \boldsymbol{B}_{i,j}^2}}. \tag{5.28}$$

In an effort to avoid notational confusion in Eq. 5.28, the projection operator is written for only the j^{th} basis component. For the reader who is familiar with the algebraic reconstruction techniques and numerically stable matrix inversion methods, we note the obvious similarity of this projection operator to Kaczmarz's iterative method for solving systems of linear equations $Ax = b$.

Any additional constraints coming from our prior information about the hyperspectral data (positivity, bounded energy, and limited support for quantized data, to name a few) can be used to further improve the results by defining new constraint sets and their corresponding projection operators accordingly. Given the projection operator above, estimates of the hi-res targets are computed iteratively from the low-res observations as follows:

$$\tilde{f}_j^{(l+1)}[n_1, n_2, k_r] = P_{C_Q(m_1,m_2,k)} \ldots P_{C_1(m_1,m_2,k)}[\tilde{f}_j^{(l)}[n_1, n_2, k_r]]. \tag{5.29}$$

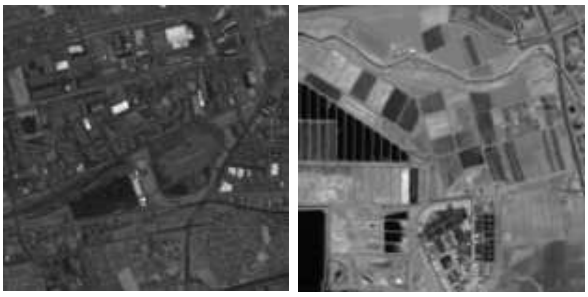
As mentioned in Section 5.1.4, this method is capable of handling different blur filters for every basis function. This is possible by modifying the projection operators in such a way that every coefficient plane is projected by using the corresponding spatial blur filter. This does not increase the computational load since the number of computations is not altered (as long as the spatial blur filters corresponding to different basis functions do not differ greatly in size).

Regardless of the method used to solve for the target image, a good share of the total effort goes into calculating the LTSV blur filter h_b. From Eq. 5.17 we see that h_b has a complex structure. It depends on the reconstruction and spatial blurring filter as well as the motion in the scene. Furthermore, h_b is only valid where the motion is accurately modeled, making the precision of the motion vectors used in the computations extremely important, [42]. In many cases the computational load renders the real-observation calculation of the generalized blur filter for every pixel impossible. Instead h_b is computed off-line and tabulated for various motion and blur values. A good understanding of the LTSV filtering operation given in Eq. 5.20 is helpful here. For this reason, in the next section we will study two special cases, namely, the case of a single observation and the case with multiple

observations and translational motion. In these cases h_b is fairly easy to compute and has a nice interpretation, which sheds some light on the LTSV filtering performed by h_b.

5.3 Experimental setup

The proposed technique is tested on a 224 band (Q=224) image of an urban area (Moffett Field) acquired by the AVIRIS hyperspectral imaging system. For detailed information on the data set see [70]. Since the original image dimensions are too large, some specific regions are cropped from the original data, and the tests are conducted on these smaller images. Figure 5.4 shows the images used in the tests. Since the AVIRIS data includes bandwidths well beyond the visible range, it is meaningless to render RGB images. For this reason a specific frequency band is selected (the hundredth band is used for all the figures in this thesis) and presented for visual purposes.



(a) AVIRIS - 1 (b) AVIRIS - 2

Figure 5.4. Hyperspectral test images

The proposed method is tested under three different motion scenarios, namely, single cube (no motion), multiple cubes with global translational motion and multiple cubes with global affine motion. Note that for the type of images we are working on, these are relevant and realistic motion models. The experimental setup can be explained as follows. In the single cube case, we begin by choosing a target window in the original hyperspectral image. Then this target window is blurred by a Gaussian filter and filtered in the spectral domain to decrease the number of spectral bands (from 31 to 15 for the first set and, from 224 to 112 for the second set). The spectrally filtered image is then downsampled in the spatial

domain to obtain the source to be used for obtaining the high resolution (both spatially and spectrally) target window. In the translational motion case, after the initial target window is chosen, we move in some predetermined direction and capture another window of the same size and we continue in this fashion until we have as many source cubes as we desire. We then proceed as we did in the single cube case to spatially blur, filter in the spectral domain and downsample the target windows. The resulting source cubes are then used to obtain the first captured high resolution target window by applying the proposed technique. The affine motion case is similar to the translational motion case except that the motion model is a six parameter affine motion model that is capable of representing rotation, scaling and translation. We have three different configurations for each scenario:

Case 1:
- 3×3 Gaussian spatial blur filter with unit variance.
- Gaussian spectral blur filter with a variance of two.
- Downsampling ratio is two in both vertical and horizontal directions.
- For the multi-cube case eight source cubes are used. translational and affine motion).

Case 2:
- 5×5 Gaussian spatial blur filter with unit variance.
- Gaussian spectral blur filter with a variance of two.
- Downsampling ratio is three in both vertical and horizontal directions.
- For the multi-cube case eight source cubes are used. translational and affine motion).

Case 3:
- 5×5 Gaussian spatial blur filter with a variance of two.
- Gaussian spectral blur filter with a variance of two.
- Downsampling ratio is three in both vertical and horizontal directions.
- For the multi-cube case fifteen source cubes are used. translational and affine motion).

The motion vectors are calculated by applying an optical flow method [71] on the properly upsampled images. Numerical results in terms of two different fidelity measures are presented in the following section. These measures are *PSNR* and *band-averaged PSNR*. In this thesis *PSNR* is defined as

$$PSNR = 10 \log_{10} \left(\frac{S_{peak}}{MSE} \right) dB \tag{5.30}$$

where S_{peak} stands for the *peak signal power*, and *band-averaged PSNR* is defined as

$$APSNR = 10\log_{10}\left(\frac{\frac{1}{Q}\sum_{i=1}^{Q} S_{peak,i}}{MSE}\right) dB \tag{5.31}$$

where $S_{peak,i}$ stands for the *peak signal power in the i^{th} spectral band*. Since the data we work on is *not* quantized, the maximum signal value is not fixed. The *band-averaged PSNR* (*APSNR*), for which the numerator is calculated as the average of the peak signal powers of all bands, is selected to compensate for this fact. Under all scenarios, the projection iterations are terminated if either the decrease in mean square error is smaller than a predetermined threshold or 5 full iterations are completed.

5.4 Simulation results

We provide the following simulation results to demonstrate the proposed method under the three scenarios mentioned above together with the results of bilinearly interpolating the separate spectral bands. Since the relevant output format depends on the intended application, both numerical and visual results will be presented. The numerical results given in Tables 5.2, and 5.3 below are *PSNR* and *APSNR* values in deciBels, where *PSNR* and *APSNR* are defined as in 5.30 and 5.31, respectively. Visual results are demonstrated in Figures 5.5, 5.6 and 5.7.

For each of the AVIRIS images, we have also included the results of comparing the proposed method with the separate-band superresolution under two different noise scenarios to differentiate between the improvement coming from projecting the additive noise to a lower dimensional space and the improvement due to the spectral de-blurring. Table 5.4, which reports numerical results in terms of APSNR values, summarizes the comparison results for translational motion case. In the noiseless case, the additive noise component is set to zero. In the noisy case, all the input images are corrupted with white Gaussian noise with a standard deviation of 50.

From the Tables 5.2, 5.3 and 5.4, we can see that the proposed method even with a single source cube performs better than bilinear interpolation. Using multiple cubes further

Table 5.2. Numerical results for AVIRIS reflectance data

AVIRIS Reflectance Data - 1				
	Bilinear interpolation	Single-cube POCS (no motion)	Multi-cube POCS (translational)	Multi-cube POCS (affine)
Case 1				
PSNR	30.7311	31.9419	34.2991	33.9251
APSNR	29.5954	30.8062	33.1635	32.8135
Case 2				
PSNR	28.1902	28.4384	31.8318	31.3857
APSNR	27.0529	27.3012	30.6931	30.2882
Case 3				
PSNR	28.0517	28.5966	31.3429	30.8498
APSNR	26.9160	27.4610	30.2073	29.7698

AVIRIS Reflectance Data - 2				
	Bilinear interpolation	Single-cube POCS (no motion)	Multi-cube POCS (translational)	Multi-cube POCS (affine)
Case 1				
PSNR	28.7236	30.1168	32.4605	32.0605
APSNR	26.9182	28.3114	30.6551	30.3161
Case 2				
PSNR	25.8142	26.1588	30.3042	29.8552
APSNR	24.0087	24.3534	28.4988	27.9837
Case 3				
PSNR	25.6275	26.4504	29.6072	26.1181
APSNR	23.8220	24.6449	27.8018	27.3544

Table 5.3. Numerical results for AVIRIS radiance data

AVIRIS Radiance Data - 1				
	Bilinear interpolation	Single-cube POCS (no motion)	Multi-cube POCS (translational)	Multi-cube POCS (affine)
Case 1				
PSNR	33.7538	34.6515	36.2611	35.8813
APSNR	29.3786	30.2763	31.8859	31.5359
Case 2				
PSNR	32.1940	32.6873	35.2154	34.7654
APSNR	28.0580	28.5499	31.0748	30.6748
Case 3				
PSNR	31.9995	32.8998	34.5228	34.0240
APSNR	27.6242	28.5246	30.1476	26.6976

AVIRIS Radiance Data - 2				
	Bilinear interpolation	Single-cube POCS (no motion)	Multi-cube POCS (translational)	Multi-cube POCS (affine)
Case 1				
PSNR	31.1782	32.2323	33.6624	33.2976
APSNR	27.1053	28.1595	29.5896	29.2396
Case 2				
PSNR	29.5590	30.2104	32.7765	32.3265
APSNR	25.7276	26.3790	28.9451	28.5451
Case 3				
PSNR	29.3244	30.5903	32.2658	31.7658
APSNR	25.2516	26.5174	28.1930	27.7430

Table 5.4. Comparison between the proposed method and applying superresolution to every band separately under no additive noise and Gaussian additive noise with a standard deviation of 50. The reported results are APSNR values, where APSNR is defined as in 5.31.

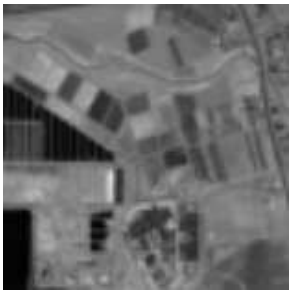
AVIRIS Reflectance Data				
	Noiseless		Noisy	
	Data set 1	Data set 2	Data set 1	Data set 2
Case 1				
Proposed method	33.6199	31.4598	33.1635	30.6551
Separate band SR	33.3188	30.8830	31.9187	29.4572
Case 2				
Proposed method	31.1186	29.0829	30.6931	28.4988
Separate band SR	30.8390	28.6340	29.8106	27.5193
Case 3				
Proposed method	30.4887	28.1946	30.2073	27.8018
Separate band SR	30.3730	27.9681	29.0673	26.5895

AVIRIS Radiance Data				
	Noiseless		Noisy	
	Data set 1	Data set 2	Data set 1	Data set 2
Case 1				
Proposed method	32.2762	30.2261	31.8859	29.5896
Separate band SR	31.9330	29.5824	31.5647	28.9270
Case 2				
Proposed method	31.3169	29.2417	31.0748	28.9451
Separate band SR	31.1164	29.0283	30.4718	28.3374
Case 3				
Proposed method	30.4338	28.6553	30.1476	28.1930
Separate band SR	30.2816	28.3514	29.3522	27.3012

improves the results, thus pointing out the advantage of fusing the information present across overlapping sources. Visual results presented in Figures 5.5, 5.6 and 5.7 also confirm the improvement seen in $PSNR$ and $APSNR$ values. The proposed model is capable of utilizing multiple bands (by projecting on each every observed spectral band separately), since the responses of the spectral blur filters are known. By exploring this knowledge in the projection operators one can achieve better results compared to applying superresolution to all bands separately, Table 5.4. The obvious reason for this is applying superresolution to the blurred spectral bands separately causes even more mixing between the bands and the additive noise components present in each band.

From Table 5.4, we can also see that for AVIRIS data, the improvement coming from the spectral de-blurring is not as much as the improvement due to noise reduction. But this is to be expected due to the characteristics of the spectral dimension of the data. AVIRIS

(a) Original



(b) Case 1: Bilinear (c) Case 1: Single cube (d) Case 1: Multi cube



(e) Case 2: Bilinear (f) Case 2: Single cube (g) Case 2: Multi cube



(h) Case 3: Bilinear (i) Case 3: Single cube (j) Case 3: Multi cube

Figure 5.5. Results for the second reflectance test image extracted from 224-band Moffett Field (AVIRIS Reflectance Data - 2). The presented multi-cube results are for translational motion scenario.

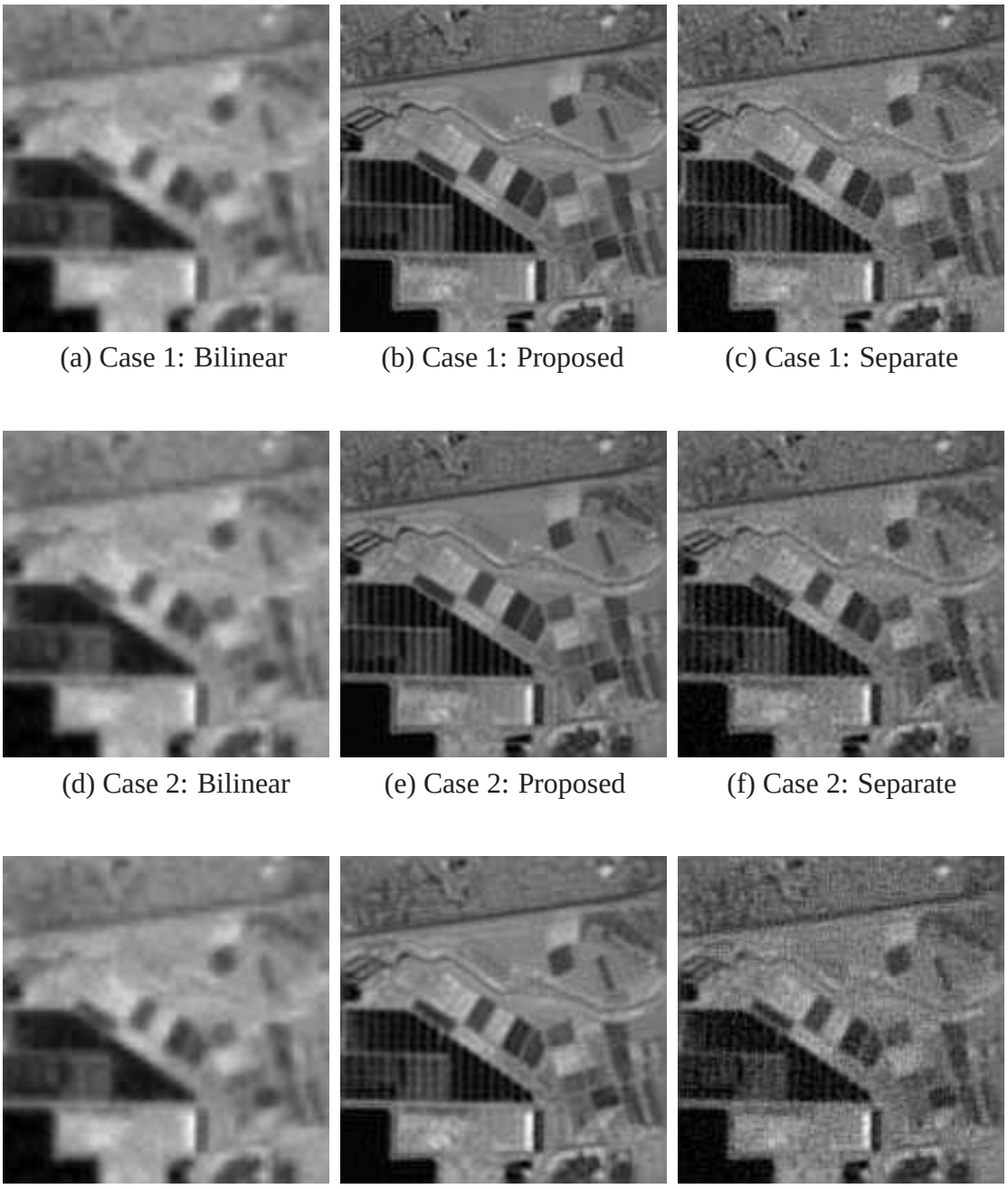
(a) Original

(b) Case 1: Bilinear (c) Case 1: Single cube (d) Case 1: Multi cube

(e) Case 2: Bilinear (f) Case 2: Single cube (g) Case 2: Multi cube

(h) Case 3: Bilinear (i) Case 3: Single cube (j) Case 3: Multi cube

Figure 5.6. Results for the second radiance test image extracted from 224-band Moffett Field (AVIRIS Radiance Data - 2). The presented multi-cube results are for translational motion scenario.



(a) Case 1: Bilinear (b) Case 1: Proposed (c) Case 1: Separate

(d) Case 2: Bilinear (e) Case 2: Proposed (f) Case 2: Separate

(g) Case 3: Bilinear (h) Case 3: Proposed (i) Case 3: Separate

Figure 5.7. Results for the second radiance test image extracted from 224-band Moffett Field (AVIRIS Radiance Data - 2). The presented multi-cube results are for translational motion scenario under Gaussian noise with a standard deviation of 50.

is a highly developed instrument capable of sampling the spectrum at more than 200 frequencies. As a result of this high number of spectral samples the observed spectrum is

usually quite smooth and the improvement over linear interpolation of the sampled spectra (for blurred and downsampled case) or doing nothing at all (for blur only case) is usually around 0.3 dB. This improvement gets larger as the spectral blurring becomes heavier. The improvement due to noise reduction is usually larger than the improvement due to spectral de-blurring and gets even larger as the noise power is increased within limitations of POCS based superresolution methods.

5.5 Material specific superresolution of hyperspectral imagery

In many hyperspectral imagery applications, dimensionality reduction through PCA is an integral component of the solution. The reason is two-fold: First, limitations on the computational budget may dictate dimensionality reduction. Second, in many applications final results are reported to a human observer who can not handle the full dimensionality of hyperspectral data, and dimensionality reduction is required to render efficient interpretation possible. Hence, integration of PCA within our resolution enhancement framework makes sense. Our proposed hyperspectral superresolution technique simultaneously reduces dimensionality, suppresses undesired noise and performs resolution enhancement on the transformed data.

But in certain applications, such as mining and petroleum exploration, we are interested in detecting specific materials with well-defined spectral signatures. Although PCA is an effective dimensionality reduction tool, it can not emphasize individual spectral signatures of interest. This is a direct result of the fact that principal components represent the highest variance (information) as a linear combination of multiple spectral signatures. For example, in military applications, such as enemy ground observation, camouflaged vehicles may not be directly recognizable if observed in the main principle component. But if we consider the bands that represent certain metal or alloys known to constitute such vehicles, detection problem is greatly alleviated. Hence, if we are interested only in a specific spectral signature, then all the other materials in the scene can be considered as interference, and

integrating PCA within our observation model does not make sense. These observations lead to the idea of material-specific hyperspectral superresolution.

5.5.1 Problem statement

Typical hyperspectral pixels are not *pure* in term of spectral content, meaning that each hyperspectral pixel is a combination of several different spectral signatures. Such pixels are called *mixed* pixels, as opposed to the pure pixels that consist of a single unique spectral signature. Mixed pixels exist for one of two reasons. First, the typical spatial resolution of hyperspectral sensors are in the scale of tens of meters. Hence, the spatial coverage of each pixel may include several different materials with different spectral signatures. Second, regardless of the spatial resolution, distinct materials can be found as homogenous mixtures. Due to this spectral mixing phenomenon, hyperspectral pixels are frequently analyzed in terms of *spectral mixing models*. Mixing models represent the acquired hyperspectral pixels as combinations of a limited number of constituent spectral endmembers, where spectral endmembers are defined as spectrally pure features such as vegetation, soil, etc. Spectral signatures of pure endmembers are usually defined under idealized laboratory conditions with controlled illumination. Although this is a perfectly valid approach to obtain spectral endmembers, it is inherently plagued by a serious shortcoming: Endmember signatures obtained in controlled laboratory conditions can not reflect the atmospheric effects present in field data. There are two alternative approaches to remedy this short coming. The first approach is based on the assumption that endmember samples can be placed/marked in the scene prior to data collection. This approach can provide true observed spectral signatures of the endmembers placed/marked in the scene. But on the other hand, placing/marking samples in the field is quite demanding and cumbersome. Furthermore, there are certain cases where it is not possible to place/mark endmember samples (due to lack of time or physical access to the scene). The second approach is to extract endmember signatures directly from the observed data. This task is referred to as *endmember extraction*, and is typically based on certain properties of the endmembers. For a detailed treatment of

endmember extraction please refer to [65].

Spectral mixing models provide the foundation for analyzing and processing hyperspectral data. Unfortunately, a complete model of spectral mixing process is way more complicated than merely describing how surface mixtures interact. A precise mixing model requires the integration of a variety physical factors including three-dimensional topology of objects in the scene, random shades, and sensor observation angle, excessively increasing complexity. To provide workable models researchers are forced to neglect these complicated physical effects. Hence, current mixing models are based on the simple assumption that within a given scene, the surface is dominated by a small number of endmembers. The fractions endmembers appear in a mixed pixel are called *fractional abundances*. Based on this assumption, observed pixels are modeled as combinations of these endmembers. Depending on how the combination mechanism is modeled, different mixing models result. The most popular mixing model, namely, the linear mixing model (LMM), assumes that the observed pixels can be represented as linear combinations of deterministic endmembers. Non-linear mixing models (NLMM) incorporate more complicated physical effects, and represent observed spectral pixels as non-linear combinations of the deterministic endmembers. Finally, the stochastic mixing model (SMM) assumes that the endmembers are distinct probability distributions whose parameters are estimated directly from the observed hyperspectral pixels. Once the endmembers are estimated, the observed pixel values are represented as linear combinations of the endmembers. Typically the endmembers are modeled as multivariate Gaussians and the resulting model is a Gaussian mixture. Finally, given a mixing model and the endmembers present in an observed scene, the task of estimating fractional abundances is referred to as *spectral unmixing*. For a detailed treatment of spectral mixing please refer to [65].

The first step in applying our hyperspectral image acquisition model to material specific super-resolution is to integrate a linear spectral mixing model into the current framework. For our purposes any linear mixing model is applicable. Once we have an observation model that relates the observed hyperspectral pixels to the endmembers and spatially aliased abundance maps, we can move to superresolution. We shall start with presenting a projection operator that optimally filters out undesired spectral signatures, and apply superresolution on the resulting projected data. The main idea is to project each hyperspectral pixel onto a subspace orthogonal to the undesired signatures. This operation can be shown to be an optimal interference suppression process in the least squares sense [72]. Once the interfering signatures have been nulled, we project the residual onto the signatures of interest and perform superresolution of the resulting image planes. This operation maximizes the SNR, and results in a small number of resolution enhanced images of materials that we are interested in.

Let us denote the spatially and spectrally continuous hyperspectral pixel with $f(x_1, x_2, \lambda, k)$. Then within the limitations of linear mixing model, we have

$$f(x_1, x_2, \lambda, k) = \sum_{j=1}^{M} e_j(\lambda) f_j(x_1, x_2, k), \tag{5.32}$$

where $e_j(\lambda)$ denotes the j^{th} endmember. Comparing Eq. 5.32 with Eq. 5.7 from Section 5.1.3, we see that the derivations of Chapter 5 are exactly applicable if we but replace the spectral PCA basis functions $b_j(\lambda)$ with the spectral endmembers $e_j(\lambda)$. Hence, the observed pixels can be represented as

$$g_i[m_1, m_2, k] = \sum_{j=1}^{P} s_{i,j} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1, n_2, k_r] h_b[m_1, m_2; n_1, n_2; k; k_r] + v[m_1, m_2, k], \tag{5.33}$$

where

$$s_{i,j} \triangleq \int_0^\infty e_j(\lambda) r_i(\lambda)\, d\lambda, \tag{5.34}$$

or in matrix form

$$\begin{bmatrix} g_1[\boldsymbol{m},k] \\ g_2[\boldsymbol{m},k] \\ \vdots \\ g_{112}[\boldsymbol{m},k] \end{bmatrix} = \begin{bmatrix} s_{1,1} & \cdots & s_{1,6} \\ s_{2,1} & \cdots & s_{2,6} \\ \vdots & \ddots & \vdots \\ s_{112,1} & \ldots & s_{112,6} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1[\boldsymbol{n},k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m};\boldsymbol{n};k;k_r] \\ \boldsymbol{f}_2[\boldsymbol{n},k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m};\boldsymbol{n};k;k_r] \\ \vdots \\ \boldsymbol{f}_6[\boldsymbol{n},k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m};\boldsymbol{n};k;k_r] \end{bmatrix} + \begin{bmatrix} v_1[\boldsymbol{m},k] \\ v_2[\boldsymbol{m},k] \\ \vdots \\ v_{112}[\boldsymbol{m},k] \end{bmatrix}$$

$$\boldsymbol{g} = \boldsymbol{S}\boldsymbol{H}\boldsymbol{f} + \boldsymbol{v}, \tag{5.35}$$

where we assume that the number of observed spectral bands is $Q = 112$, the number of endmembers is $P = 6$, and use the following definitions to simplify the expression:

$$\boldsymbol{f}_j[\boldsymbol{n},k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m};\boldsymbol{n};k;k_r] \triangleq \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_j[n_1,n_2,k_r] h_b[m_1,m_2;n_1,n_2;k;k_r], \tag{5.36}$$

$$\boldsymbol{f}[\boldsymbol{n},k_r] \triangleq \begin{bmatrix} \boldsymbol{f}_1[\boldsymbol{n},k_r] \\ \boldsymbol{f}_2[\boldsymbol{n},k_r] \\ \vdots \\ \boldsymbol{f}_6[\boldsymbol{n},k_r] \end{bmatrix}, \quad \boldsymbol{H}\boldsymbol{f}[\boldsymbol{m},k] \triangleq \begin{bmatrix} \boldsymbol{f}_1[\boldsymbol{n},k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m};\boldsymbol{n};k;k_r] \\ \boldsymbol{f}_2[\boldsymbol{n},k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m};\boldsymbol{n};k;k_r] \\ \vdots \\ \boldsymbol{f}_6[\boldsymbol{n},k_r] \cdot \boldsymbol{h}_b[\boldsymbol{m};\boldsymbol{n};k;k_r] \end{bmatrix},$$

$$\boldsymbol{g}[\boldsymbol{m},k] \triangleq \begin{bmatrix} g_1[\boldsymbol{m},k] \\ g_2[\boldsymbol{m},k] \\ \vdots \\ g_{112}[\boldsymbol{m},k] \end{bmatrix}, \quad \boldsymbol{S} \triangleq \begin{bmatrix} s_{1,1} & \cdots & s_{1,6} \\ s_{2,1} & \cdots & s_{2,6} \\ \vdots & \ddots & \vdots \\ s_{112,1} & \ldots & s_{112,6} \end{bmatrix}.$$

The similarity between inverting Eq. 5.35 and the spectral unmixing problem is worth noting. If we choose the downsampling ratio as one, that is, if we assume the target and observed images are at the same spatial resolution, then the resulting problem is equivalent to spectral unmixing with multiple registered observations. The main advantage of such an approach would be the increased robustness against observation noise provided by multiple observations for each pixel location (as a result of spatial registration). When the target image is of higher resolution, the resulting problem is similar to the problem described in Section 5.2. By using a POCS based iterative inversion algorithm similar to the algorithm

outlined in Section 5.2, we could obtain a superresolved abundance map that would be valuable for subpixel target detection. However, we will concentrate on superresolving the abundance map of a single endmember by incorporating an additional step into our imaging model.

5.5.2 Least-squares optimal projection operator

Without losing any generality, let us assume that we are interested on a single endmember. All the derivations to follows can be easily extended to multiple endmembers. Superresolving a specific endmember is complicated by the correlation between endmembers and the presence of noise. A direct correlation-based approach by projecting the observed pixels on the endmember of interest is suboptimal [73]. To see this, note that along with noise, all the endmembers we are not interested act as interference, and the correlation between the target endmember and these undesired endmembers can be quite high, at least in certain spectral bands. We can achieve better results if we project the observed hyperspectral pixels onto a custom designed subspace.

Let us start with the effects of the interfering spectral signatures. Note that the columns of the $\boldsymbol{S}$ matrix are the endmembers after the application of spectral filtering. We can rearrange columns of $\boldsymbol{S}$ so that the endmember of interest is the first column. Then we have

$$\boldsymbol{S} = [\boldsymbol{d} \quad \boldsymbol{U}], \tag{5.37}$$

where the column vector $\boldsymbol{d}$ is the desired endmember and $\boldsymbol{U}$ is the matrix consisting of interfering endmembers. To eliminate the effects of the interfering endmembers, we will apply the technique proposed in [72]. The main idea is to project the observed hyperspectral signature onto the a subspace that is orthogonal to all interfering spectral signatures. This is equivalent to projecting onto the nullspace of $\boldsymbol{U}^T$. We use a classic result from linear algebra to write the least squares optimal projection operator as

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^{\dagger}, \tag{5.38}$$

where $\boldsymbol{U}^{\dagger}$ denotes the pseudoinverse of $\boldsymbol{U}$. If $\boldsymbol{U}$ is full-column rank then $\boldsymbol{U}^{\dagger} = (\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T$. The resulting vector will only have energy coming from the desired spectral signature and noise. The application of $\boldsymbol{P}$ effectively removes contribution from the columns of $\boldsymbol{U}$, that is,

$$\boldsymbol{P}\boldsymbol{g} = \boldsymbol{P}\boldsymbol{d}\tilde{f}_d + \boldsymbol{P}\boldsymbol{v}, \tag{5.39}$$

where $\tilde{f}_d$ is the element of the vector $\boldsymbol{H}\boldsymbol{f}$ that corresponds to the desired endmember. The next step is to minimize the effect of remaining noise component. Let us denote the operator that maximizes SNR with $\boldsymbol{q}^T$. Then we have

$$\boldsymbol{q}^T\boldsymbol{P}\boldsymbol{g} = \boldsymbol{q}^T\boldsymbol{P}\boldsymbol{d}\tilde{f}_d + \boldsymbol{q}^T\boldsymbol{P}\boldsymbol{v}. \tag{5.40}$$

Then the ratio of signal energy to noise energy is given as

$$\lambda = \frac{\boldsymbol{q}^T\boldsymbol{P}\boldsymbol{d}\tilde{f}_d^2\boldsymbol{d}^T\boldsymbol{P}^T\boldsymbol{q}}{\boldsymbol{q}^T\boldsymbol{P}\mathcal{E}[\boldsymbol{v}\boldsymbol{v}^{\mathcal{T}}]\boldsymbol{P}^{\mathcal{T}}\boldsymbol{q}} = \frac{\alpha_d^2}{\sigma^2}\frac{\boldsymbol{q}^T\boldsymbol{P}\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{P}^T\boldsymbol{q}}{\boldsymbol{q}^T\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{q}}, \tag{5.41}$$

where $\mathcal{E}[\cdot]$ denotes statistical expectation. The ratio in Eq. 5.41 is in a form known as the Rayleigh quotient, and its maximization is the well-studied generalized eigenvector problem [74] that can be stated as

$$\boldsymbol{P}\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{P}^T\boldsymbol{q} = \lambda\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{q}. \tag{5.42}$$

Noting $\boldsymbol{P}$ is symmetric ($\boldsymbol{P} = \boldsymbol{P}^T$) and idempotent ($\boldsymbol{P}^2 = \boldsymbol{P}$), the solution is given as

$$\boldsymbol{q}^T = \kappa\boldsymbol{d}^T, \tag{5.43}$$

where κ is an arbitrary constant [3]. Finally, the combination of these two projection operators gives the least squares optimal projection operator we desire, that is,

$$\boldsymbol{Q} = \boldsymbol{q}^T\boldsymbol{P} = \kappa\boldsymbol{d}^T\boldsymbol{P}. \tag{5.44}$$

For a detailed treatment of the derivation (for single and multiple spectral signatures of interest) and computation of the projection operator please refer to [72] and [75].

[3] Note that this is the well-known matched filter.

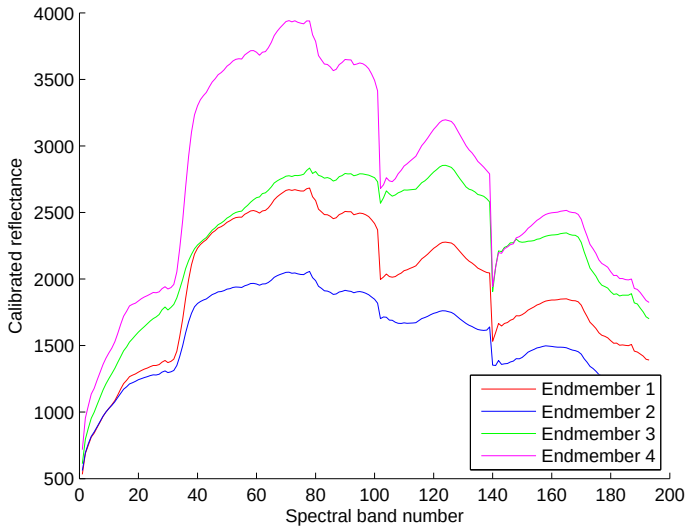
From Eq. 5.44 and Eq. 5.40 we can see that upon application of the proposed projection operator we are left with the conventional superresolution setup. We have a number of warped, aliased and noisy single-plane abundance maps of a predetermined endmember, and our goal is to obtain a superresolved single-plane abundance map. This inverse problem can be solved with any of the superresolution algorithms proposed in [67], [29], and [42]. We preferred to use the POCS based technique detailed in [42].

5.6 Experimental setup and simulation results

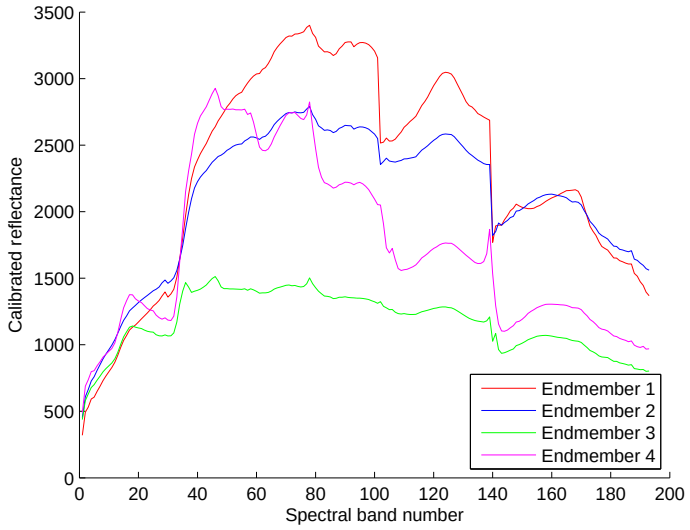
To test the proposed method we require ground truth hyperspectral data with a complete set of endmembers and corresponding fractional abundances. Such ground truth data is very hard, if possible, to obtain. We could *generate* random spectral signatures or pick spectral signatures from existing endmember libraries, and synthesize pixels by linearly combining these signatures with random weights. This is a feasible approach for obtaining numeric results, but it becomes very cumbersome if we want to provide visual results with meaningful spatial structure. To get around this obstacle we used the following approach. We started with the AVIRIS reflectance data set used in Section 5.3. We first applied the technique proposed in [76] to extract the endmembers shown in Figure 5.8. Then assuming the linear mixing model, we applied the nonnegative least squares (NNLS) method summarized in [65] to estimate the fractional abundance maps for all endmembers. Since the extracted endmembers can never perfectly match the true endmembers in the scene [4] the obtained fractional abundance maps can not represent the data with zero (or negligibly small) error. To have a scene with perfectly matching endmembers and abundance maps, we synthesized a new hyperspectral data set using the computed fractional abundance maps and the extracted spectral endmembers under the linear mixing model. All experiments presented in this section are conducted on this synthesized data. Note that the validity of our results

[4] We do not even know the *true* number of endmembers present in the scene. Furthermore, the data calibration process and the observation noise distort the data and the linearity assumption of linear mixing model.

is not dependent on the specific endmember extraction technique used. For all practical purposes, we could have even assumed that the endmembers were given. But for the data set we picked, such predetermined endmembers and corresponding abundance maps were not available at the time we conducted our experiments. Hence, we extracted our own endmembers, and estimated the corresponding abundance maps.



(a) Reflectance - 1



(b) Reflectance - 2

Figure 5.8. Endmembers extracted from AVIRIS reflectance data

For the visual results demonstrated in Figures 5.9 and 5.10 the fourth endmember shown

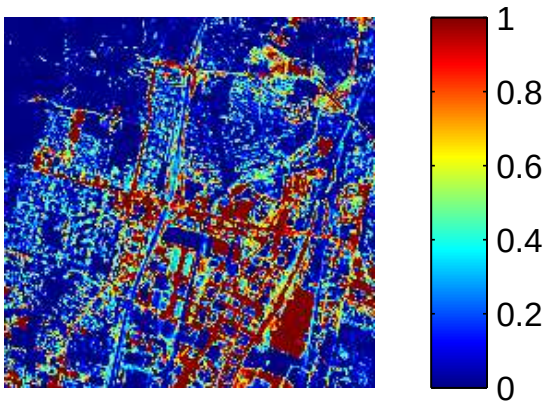
Table 5.5. Numerical results for AVIRIS reflectance data with and without noise

AVIRIS Reflectance Data - 1		
	Bilinear interpolation	Proposed method
Case 1 (Noise free)	20.03	26.76
Case 2 (AWGN $\sigma = 30$)	18.55	22.73

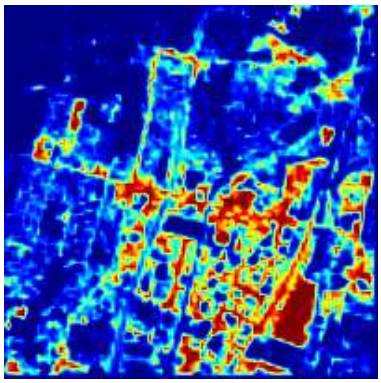
AVIRIS Reflectance Data - 2		
	Bilinear interpolation	Proposed method
Case 1 (Noise free)	21.07	26.71
Case 2 (AWGN $\sigma = 30$)	19.77	21.34

in Figure 5.8 is used. The optimal subspace projection based superresolution method is compared to bilinearly interpolating the abundance map obtained by applying the proposed projection operator on a single low resolution observation under global translational motion scenario. To simulate global translational motion we shift a selected window in some predetermined direction and capture another window of the same size. We continue in this fashion until we have as many source cubes as we desire. The shifted windows are spatially blurred with a Gaussian blur filter with unit variance and downsampled by three in both spatial dimensions. For spatial noise we experimented with two cases, namely, noise free (Case 1) and additive white Gaussian noise (AWGN) with a standard deviation of 30 (Case 2). For the sake of simplicity we ignored spectral blurring effects.

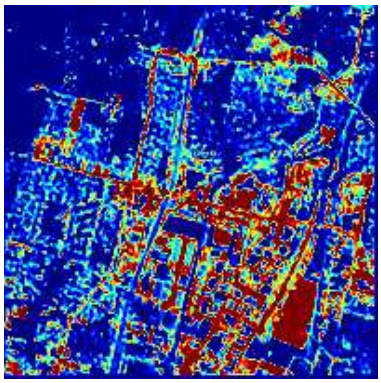
Numerical results in terms of $PSNR$ as defined in Eq. 5.30 are given in Table 5.5. From the Table 5.5, we can see that the proposed method clearly outperforms bilinear interpolation. Visual results presented in Figures 5.9 and 5.10 also confirm the improvement seen in $PSNR$ values. The proposed method effectively suppresses the effects of interfering spectral signatures and noise, and achieves spatial resolution enhancement.



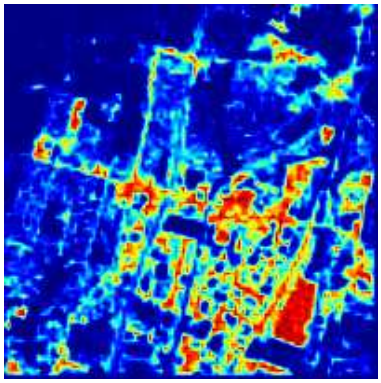
(a) Original



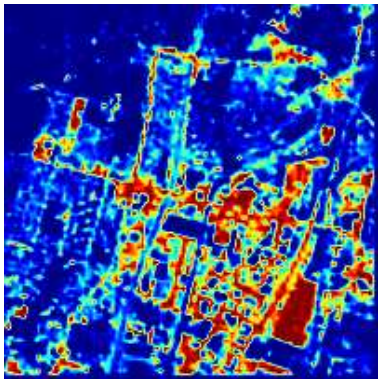
(b) Case 1 - Bilinear



(c) Case 1 - Proposed

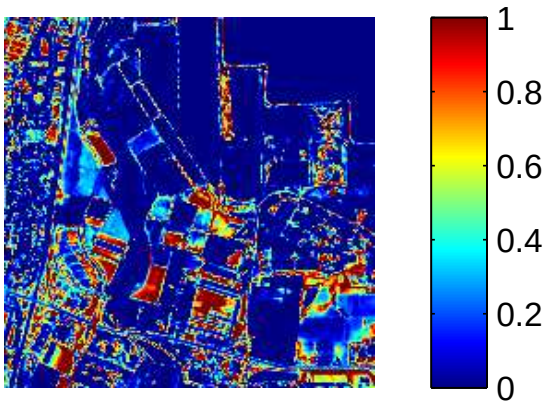


(d) Case 2 - Bilinear

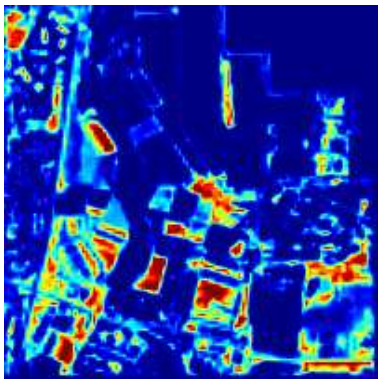


(e) Case 2 - Proposed

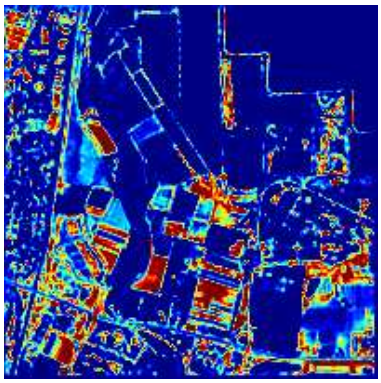
Figure 5.9. Results for the first reflectance test image extracted from 224-band Moffett Field (AVIRIS Reflectance Data - 1).



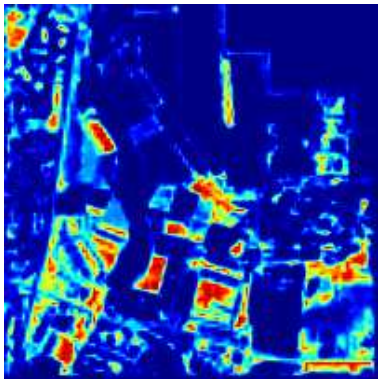
(a) Original



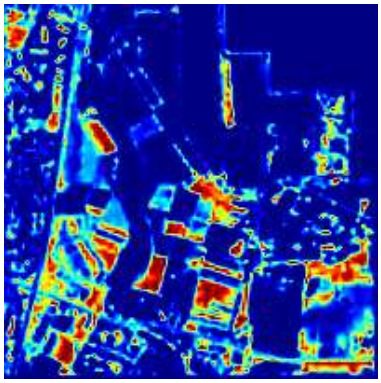
(b) Case 1 - Bilinear



(c) Case 1 - Proposed



(d) Case 2 - Bilinear



(e) Case 2 - Proposed

Figure 5.10. Results for the second reflectance test image extracted from 224-band Moffett Field (AVIRIS Reflectance Data - 2).

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis we focused on single and multi-frame spatial resolution enhancement with applications to resolution standards conversion and hyperspectral imagery. We conclude with a discussion of the assumptions we made during derivations of the proposed algorithms, limitations introduced by these assumptions, and future research directions that could further improve and generalize our work.

In single-frame resolution enhancement our main assumption is that the input image/video signal is natural looking. There are certain cases of practical importance that violate this assumption. First, consider the scaling of highly structured signals such as monoscope images[1] and superimposed text such as subtitles in DVD movies. These signals exhibit quite different spatial structures compared to natural images, which is a potential problem for training based resolution enhancement approach. Training based methods, such as resolution synthesis, are designed to capture the characteristics of the data set over which they were trained. If our training set consists of only natural scenes without any representatives of monoscope images and text, the resulting context classes and interpolation filters can not be expected to perform satisfactorily for these images. If we include these signals in the training set, then we face another problem. Due to their spatial characteristics, their context prototypes and optimal interpolation filters are different compared to natural signals. Hence, by training on a set of mixed signals we try to adapt to two completely different types of signals. The resulting context classes and filters are not truly fitted to any signal type. This points out the need to spatially differentiate different types of input signals, and apply appropriately trained context classes and filters. Signal classification can be integrated into the pixel classification block through the use of modified features that allow us to differentiate text from natural content.

[1] Monoscope images are used as test signals in TV broadcasting, and they typically consist of artificially produced patterns of varying spatial frequency and sharp color transitions.

To avoid complications caused by compression artifacts, we developed and tested our algorithms on DVD (or higher) resolution image/video sequences with very low to no compression. Enhancement of low bit-rate image/video signals is a challenge for both single and multi-frame techniques. Accurate sub-pixel motion estimation on low bit-rate video is a very challenging problem. Currently, most of the motion estimation algorithms are block based. Loss of high frequency detail and presence of blocking artifacts cause serious performance degradations for block based motion estimation methods. Furthermore, at low enough bit-rates, even medium range frequency components can be damaged or lost completely, and feature extraction required by single-frame techniques becomes very challenging.

Finally, we designed and implemented resolution enhancement as a stand alone processing block with the only exception being possible denoise filtering prior to resolution enhancement. Unfortunately, in real signal processing systems, this is rarely the case. In practice, image/video signals are processed by pipelines of blocks such as compression/decompression, denoising, sharpening, contrast enhancement, color/spectral enhancement, and gamma correction. The interaction of resolution enhancement with these processing blocks is an interesting research direction for two main reasons. First, typically scaling/resolution enhancement is among the computationally most demanding processing blocks. Hence, any improvement offered by resolution enhancement is truly hard-earned. If any other processing block on the pipeline has the potential to introduce modifications that can suppress or even destroy this improvement, then its effects and optimal placement on the pipeline should be carefully studied. Second, by taking the other processing blocks into account we should be able to achieve better and more visible resolution enhancement.

APPENDIX A

TELEVISION INDUSTRY TERMS AND DEFINITIONS

This Appendix aims to explain the terms related to the television industry and standards.

Flicker problem: Flicker is the visible fading between frames displayed on a CRT display. CRT displays are based on hitting a phosphor-coated screen with a focused and properly deflected electron beam. Every phosphor element corresponds to a pixel and needs to refreshed at a fixed rate by continuously sweeping the screen with the electron gun. Flicker occurs when the CRT display is driven at a low refresh rate, allowing the screen's phosphors to lose their excitation between sweeps of the electron gun. Since flat panel displays use active pixels (a transistor for each pixel keeps the pixel at its desired state) they typically do not have flicker problem.

Native resolution: Native resolution is the fixed number of rows and columns of pixels offered by a flat panel display.

Standard definition: Standard definition (SD) resolution typically refers to 480×720 (480 lines and 720 columns) (interlaced) scanned at 60 Hz for NTSC standard of North America and Japan, and 576×720 (480 lines and 720 columns) (interlaced) scanned at 50 Hz for PAL/SECAM standard of Europe.

High definition: High definition (HD) resolution typically refers to 720×1280 (progressive) scanned at 60 Hz or 1080×1920 (interlaced) scanned at 50 Hz. HD offers a much better picture quality compared to SD.

Progressive scan: Progressive scan video signals are captured, transmitted and displayed as complete image frames line by line, from top to bottom.

Interlaced scan: Interlacing is an old technique to improve picture and motion quality of the TV signals without consuming additional bandwidth. Interlacing is a trade-off between spatial and temporal resolution. To obtain interlaced scan, every picture frame is

divided into two fields called even and odd fields. Even field consists of the even numbered lines of the frame, while the odd field consists of the odd numbered lines. Instead of transmitting the whole picture, interlaced scan transmits only the even or odd field of every frame in an alternating fashion. The afterglow of the phosphor of CRT tubes, in combination with the persistence of vision results in two fields being perceived as a continuous image which allows the viewing of full horizontal detail with half the bandwidth which would be required for a full progressive scan while maintaining the necessary CRT refresh rate to prevent flicker.

For fixed bandwidth and spatial resolution, interlaced scan can offer twice the refresh rate compared to progressive scan. For fixed bandwidth and refresh rate, interlaced scan can offer twice the number of pixels compared to progressive scan. But interlacing is not without short-comings. It can cause flicker and various kinds of distortions around object boundaries, especially in motion intensive scenes. Interlacing is still in use for most SD TVs, and the 1080×1920 interlaced HDTV broadcast standard. Unfortunately LCD and plasma displays are inherently progressive scan and require some form of de-interlacing.

APPENDIX B

DERIVATION OF THE OPTIMAL FILTER COEFFICIENTS

The optimal filter coefficients are the solutions of the following optimization problem:

$$\min_{\boldsymbol{A}_j,\,\boldsymbol{\beta}_j}\left(\sum_{s\in S} w_{j,\boldsymbol{y}_s}\|\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\|^2\right) \tag{B.1}$$

We start with writing the cost function of Eq. B.1 as

$$\sum_{s\in S} w_{j,\boldsymbol{y}_s}\|\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\|^2=\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left[\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\right]^T\left[\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\right] \tag{B.2}$$

First we take the gradient of Eq B.2 with respect to $\boldsymbol{\beta}_j$ to obtain

$$\begin{aligned}\frac{\partial}{\partial\boldsymbol{\beta}_j}\sum_{s\in S} w_{j,\boldsymbol{y}_s}\|\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\|^2&=\sum_{s\in S} w_{j,\boldsymbol{y}_s}\frac{\partial}{\partial\boldsymbol{\beta}_j}\left\{\left[\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\right]^T\left[\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\right]\right\}\\&=\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(-2\boldsymbol{f}_s+2\boldsymbol{A}_j\boldsymbol{g}_s+2\boldsymbol{\beta}_j\right),\end{aligned} \tag{B.3}$$

where we used the following gradient formulas [77]

$$\begin{aligned}\frac{\partial\boldsymbol{x}^T\boldsymbol{y}}{\partial\boldsymbol{x}}&=\frac{\partial\boldsymbol{y}^T\boldsymbol{x}}{\partial\boldsymbol{x}}=\boldsymbol{y},\\\frac{\partial\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\partial\boldsymbol{x}}&=\left(\boldsymbol{A}+\boldsymbol{A}^T\right)\boldsymbol{x}.\end{aligned}$$

Setting Eq. B.3 to zero gives

$$\boldsymbol{\beta}_j=\sum_{s\in S} w_{j,\boldsymbol{y}_s}\boldsymbol{f}_s-\boldsymbol{A}_j\sum_{s\in S} w_{j,\boldsymbol{y}_s}\boldsymbol{g}_s. \tag{B.4}$$

We define $\overline{\boldsymbol{f}}$ and $\overline{\boldsymbol{g}}$.

$$\begin{aligned}\overline{\boldsymbol{f}}_j&=\sum_{s\in S} w_{j,\boldsymbol{y}_s}\boldsymbol{f}_s,\\\overline{\boldsymbol{g}}_j&=\sum_{s\in S} w_{j,\boldsymbol{y}_s}\boldsymbol{g}_s.\end{aligned}$$

Then we can write Eq. B.4 as

$$\boldsymbol{\beta}_j=\overline{\boldsymbol{f}}_j-\boldsymbol{A}_j\overline{\boldsymbol{g}}_j. \tag{B.5}$$

Next we take the gradient of Eq B.2 with respect to $\boldsymbol{A}_j$ to obtain

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{A}_j}\sum_{s\in S} w_{j,\boldsymbol{y}_s}\|\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\|^2 &= \sum_{s\in S} w_{j,\boldsymbol{y}_s}\frac{\partial}{\partial \boldsymbol{A}_j}\left\{\left[\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\right]^T\left[\boldsymbol{f}_s-\boldsymbol{A}_j\boldsymbol{g}_s+\boldsymbol{\beta}_j\right]\right\}\\ &= \sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(-2\boldsymbol{f}_s\boldsymbol{g}_s^T+2\boldsymbol{\beta}_j\boldsymbol{g}_s^T+2\boldsymbol{A}_j\boldsymbol{g}_s\boldsymbol{g}_s^T\right), \qquad \text{(B.6)}\end{aligned}$$

where we used the following gradient formulas [77]

$$\begin{aligned}\frac{\partial \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y}}{\partial \boldsymbol{A}} &= \boldsymbol{x}\boldsymbol{y}^T,\\ \frac{\partial \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{A}} &= \boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^T.\end{aligned}$$

Setting Eq. B.6 to zero gives

$$\boldsymbol{A}_j\sum_{s\in S} w_{j,\boldsymbol{y}_s}\boldsymbol{g}_s\boldsymbol{g}_s^T = \sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\boldsymbol{\beta}_j\right)\boldsymbol{g}_s^T. \qquad \text{(B.7)}$$

We substitute $\boldsymbol{\beta}_j$ from Eq.B.5 to obtain

$$\begin{aligned}\boldsymbol{A}_j\sum_{s\in S} w_{j,\boldsymbol{y}_s}\boldsymbol{g}_s\boldsymbol{g}_s^T &= \sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\overline{\boldsymbol{f}}_j+\boldsymbol{A}_j\overline{\boldsymbol{g}}_j\right)\boldsymbol{g}_s^T\\ \boldsymbol{A}_j\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left[\boldsymbol{g}_s\boldsymbol{g}_s^T-\overline{\boldsymbol{g}}_j\boldsymbol{g}_s^T\right] &= \sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\overline{\boldsymbol{f}}_j\right)\boldsymbol{g}_s^T\\ \boldsymbol{A}_j\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{g}_s-\overline{\boldsymbol{g}}_j\right)\boldsymbol{g}_s^T &= \sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\overline{\boldsymbol{f}}_j\right)\boldsymbol{g}_s^T. \qquad \text{(B.8)}\end{aligned}$$

Consider the following equalities

$$\begin{aligned}\boldsymbol{A}_j\left[\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{g}_s-\overline{\boldsymbol{g}}_j\right)\overline{\boldsymbol{g}}_j^T\right] &= \boldsymbol{A}_j\left[\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{g}_s-\overline{\boldsymbol{g}}_j\right)\right]\overline{\boldsymbol{g}}_j^T = 0 \qquad \text{(B.9)}\\ \sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\overline{\boldsymbol{f}}_j\right)\overline{\boldsymbol{g}}_j^T &= \left[\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\overline{\boldsymbol{f}}_j\right)\right]\overline{\boldsymbol{g}}_j^T = 0 \qquad \text{(B.10)}\end{aligned}$$

Hence we can add these terms to the left and righthand sides of Eq.B.8, respectively, to obtain

$$\begin{aligned}\boldsymbol{A}_j\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{g}_s-\overline{\boldsymbol{g}}_j\right)\boldsymbol{g}_s^T-\boldsymbol{A}_j\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{g}_s-\overline{\boldsymbol{g}}_j\right)\overline{\boldsymbol{g}}_j^T &= \sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\overline{\boldsymbol{f}}_j\right)\boldsymbol{g}_s^T-\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\overline{\boldsymbol{f}}_j\right)\overline{\boldsymbol{g}}_j^T\\ \boldsymbol{A}_j\sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{g}_s-\overline{\boldsymbol{g}}_j\right)\left(\boldsymbol{g}_s^T-\overline{\boldsymbol{g}}_j^T\right) &= \sum_{s\in S} w_{j,\boldsymbol{y}_s}\left(\boldsymbol{f}_s-\overline{\boldsymbol{f}}_j\right)\left(\boldsymbol{g}_s^T-\overline{\boldsymbol{g}}_j^T\right) \qquad \text{(B.11)}\end{aligned}$$

Finally we observe

$$\begin{aligned}
\boldsymbol{\nu}_{f|j} &= \overline{\boldsymbol{f}}_j, \\
\boldsymbol{\nu}_{g|j} &= \overline{\boldsymbol{g}}_j, \\
\boldsymbol{\Gamma}_{fg|j} &= \sum_{s \in S} w_{j,\mathbf{y}_s} \left(\boldsymbol{f}_s - \overline{\boldsymbol{f}}_j \right) \left(\boldsymbol{g}_s^T - \overline{\boldsymbol{g}}_j^T \right), \\
\boldsymbol{\Gamma}_{gg|j} &= \sum_{s \in S} w_{j,\mathbf{y}_s} \left(\boldsymbol{g}_s - \overline{\boldsymbol{g}}_j \right) \left(\boldsymbol{g}_s^T - \overline{\boldsymbol{g}}_j^T \right).
\end{aligned}$$

which imply that the optimal filter coefficients obtained from the alternate formulation are identical to the ones presented in the original derivation.

$$\begin{aligned}
\boldsymbol{A}_j &= \boldsymbol{\Gamma}_{fg|j} \boldsymbol{\Gamma}_{gg|j}^{-1}, \\
\boldsymbol{\beta}_j &= \boldsymbol{\nu}_{f|j} - \boldsymbol{\Gamma}_{fg|j} \boldsymbol{\Gamma}_{gg|j}^{-1} \boldsymbol{\nu}_{g|j}.
\end{aligned}$$

REFERENCES

[1] C. Atkins, C. Bouman, and J. Allebach, "Optimal image scaling using pixel classification," in *IEEE Proceedings of the International Conference on Image Processing, 2001*, vol. 3, pp. 864–867, Oct 2001.

[2] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, pp. 56–65, March 2002.

[3] G. Shaw and D. Manolakis, "Signal processing for hyperspectral image exploitation," *IEEE Signal Processing Magazine*, vol. 19, pp. 12–16, Jan. 2002.

[4] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Processing Magazine*, vol. 19, pp. 17–28, Jan. 2002.

[5] M. Eismann and R. Hardie, "Application of the stochastic mixing model to hyperspectral resolution enhancement," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, pp. 1924–1933, Sep 2004.

[6] R. Hardie, M. Eismann, and G. Wilson, "Map estimation for hyperspectral image resolution enhancement using an auxiliary sensor," *IEEE Transactions on Image Processing*, vol. 13, pp. 1174–1184, Sep 2004.

[7] M. Eismann and R. Hardie, "Hyperspectral resolution enhancement using high-resolution multispectral imagery with arbitrary response functions," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, pp. 455–465, Mar 2005.

[8] J. Schuler and D. Schribner, "Increasing spatial resolution through temporal supersampling of digital video," *Optical Engineering*, vol. 38, pp. 801–805, May 1999.

[9] E. P. Simoncelli, "Statistical models for images: Compression, restoration and synthesis," in *31st Asilomar Conf. on Sig., Sys. and Computers*, vol. 3071, pp. 47–60, October 1997.

[10] D. J. Field, "What is the goal of sensory coding?," *Neural Computation*, vol. 6, pp. 559–601, 1994.

[11] T. Huang and R. Tsai, *Multiframe Image Restoration and Registration, In: Advances in Computer Vision and Image Processing*, vol. 1. Greenwich, CT: JAI Press Inc, 1984.

[12] B. Tom and A. Katsaggelos, "Reconstruction of a high resolution image by simultaneous registration, restoration and interpolation of low resolution images," in *Proc. Int. Conf. Image Processing*, vol. 2, pp. 539–542, 1995.

[13] H. Ur and D. Gross, "Improved resolution from subpixel shifted pictures," *CVGIP: Graph. Models Image Processing*, vol. 54, pp. 181–186, Mar. 1992.

[14] T. Komatsu, K. Aizawa, T. Igarashi, and T. Saito, "Signal-processing based method for acquiring very high resolution images with multiple cameras and its theoretical analysis," in *IEE Proceedings on Communications, Speech and Vision*, vol. 140, pp. 19–24, Feb 1993.

[15] M. Alam, J. Bognar, R. Hardie, and B. Yasuda, "Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased video frames," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, pp. 915–923, Oct 2000.

[16] A. Shah, N.R.; Zakhor, "Resolution enhancement of color video sequences," *Image Processing, IEEE Transactions on*, vol. 8, pp. 879–885, Jun 1999.

[17] R. Y. Tsai and T. S. Huang, *Multiframe Image Restoration and Registration, In: Advances in Computer Vision and Image Processing, ed. T.S.Huang. Greenwich, CT*. JAI Press, 1984.

[18] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1013–1027, June 1990.

[19] S. Kim and W. Su, "Recursive high-resolution reconstruction of blurred multiframe images," *IEEE Transactions on Image Processing*, vol. 2, pp. 534–539, Oct. 1993.

[20] B. Tom, A. Katsaggelos, and N. Galatsanos, "Reconstruction of a high resolution image from registration and restoration of low resolution images," in *Proc. IEEE Int. Conf. Image Processing*, (Austin, TX), pp. 13–16, Nov. 1994.

[21] B. C. Tom, K. T. Lay, and A. K. Katsaggelos, "Multi-channel image identification and restoration using the expectation-maximization algorithm," *Optical Engineering, Special Issue on Visual Communications and Image Processing*, vol. 35, pp. 241–254, January 1996.

[22] M. Hong, M. Kang, and A. Katsaggelos, "An iterative weighted regularized algorithm for improving the resolution of video sequences," in *IEEE Proceedings of the International Conference on Image Processing*, vol. 2, pp. 474–477, Oct 1997.

[23] R. Hardie, K. Barnard, J. Bognar, E. Armstrong, and E. Watson, "High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging sytem," *Opt. Eng.*, vol. 37, pp. 247–260, Jan 1998.

[24] N. Bose, S. Lertrattanapanich, and J. Koo, "Advances in superresolution using l-curve," in *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 433–436, May 2001.

[25] H. Stark and P. Oskoui, "High-resolution image recovery from image-plane arrays, using convex projections," *Journal of Optical Society of America, A: Optics and Image Science*, vol. 6, pp. 1715–1726, Novomber 1989.

[26] A.J.Patti, M.I.Sezan, and A.M.Tekalp, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," *IEEE Trans. Image Processing*, vol. 6, pp. 1064–1076, August 1997.

[27] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 169–172, March 1992.

[28] A. J. Patti and Y. Altunbasak, "Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants," *IEEE Transactions on Image Processing*, vol. 10, pp. 179–186, January 2001.

[29] R. Schultz and R. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing*, vol. 5, pp. 996–1011, June 1996.

[30] R. L. Stevenson, B. E. Schmitz, and E. J. Delp, "Discontinuity preserving regularization of inverse visual problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, pp. 455–469, 1994.

[31] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy and undersampled measured images," *IEEE Transactions on Image Processing*, vol. 6, pp. 1646–1658, December 1997.

[32] T. E. Boult, M.-C. Chiang, and R. J. Micheals, *Super-Resolution via Image Warping*. Boston: Kluwer Academic Publishers, 2001.

[33] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. H. III, and R. M. Mersereau, "Eigenface-domain super-resolution for face recognition," *IEEE Transactions on Image Processing*, vol. 12, pp. 597 – 606, 2003.

[34] S. Baker and T. Kanade, "Hallucinating faces," in *Fourth International Conf. Automatic Face and Gesture Recognition*, March 2000.

[35] C. Liu, H.-Y. Shum, and C.-S. Zhang, "A two-step approach to hallucinating faces: Global parametric model and local nonparametric model," in *IEEE Conference on Computer Vision and Pattern Recognition*, December 2001.

[36] D. Capel and A. Zisserman, "Super-resolution from multiple views using learnt image models," in *IEEE Conference on Computer Vision and Pattern Recognition*, December 2001.

[37] T. Gotoh and M. Okutomi, "Direct super-resolution and registration using raw CFA images," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, vol. 2, pp. 600 – 607, 2004.

[38] A. Galbraith, J. Theiler, K. Thome, and R. Ziolkowski, "Resolution enhancement of multilook imagery for the multispectral thermal imager," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, pp. 1964–1977, Sep 2005.

[39] T. Akgun, Y. Altunbasak, and R. M. Mersereau, "Super-resolution reconstruction of hyperspectral images," *IEEE Transactions on Image Processing*, vol. 14, pp. 1860–1875, Nov 2005.

[40] D. Chen and R. R. Schultz, "Extraction of high-resolution frames from mpeg image sequences," in *IEEE International Conference on Image Processing*, (Chicago, IL), October 1998.

[41] J. Mateos, A. Katsaggelos, and R. Molina, "Resolution enhancement of compressed low resolution video," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Istanbul, Turkey), June 2000.

[42] Y. Altunbasak, A. Patti, and R. Mersereau, "Super-resolution still and video reconstruction from MPEG-coded video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 217–226, 2002.

[43] N. Nguyen and P. Milanfar, "An efficient wavelet-based algorithm for image superresolution," in *IEEE Int. Conf. Image Proc.*, vol. 2, pp. 351–354, September 2000.

[44] N. K. Bose, S. Lertrattanapanich, and M. B. Chappalli, "Superresolution with second generation wavelets," *Signal Processing: Image Comm.*, vol. 19, no. 5, p. 387391, 2004.

[45] M. Chappalli and N. Bose, "Simultaneous noise filtering and super-resolution with second-generation wavelets," *Signal Processing Letters*, vol. 12, pp. 772–775, Jan 2005.

[46] R. Willett, I. Jermyn, R. Nowak, and J. Zerubia, "Wavelet-based superresolution in astronomy," in *Proceedings of Astronomical Data Analysis Software and Systems (ADASS) XIII*, vol. 314, (Strasbourg, France), p. 107, Oct 2004.

[47] B. Uyar, M. Sayinta, T. Akgun, B. Orencik, and Y. Altunbasak, "Spatial feature based video scaling and its fpga implementation for video standards conversion," in *IEEE Workshop on Signal Processing Systems (SiPS 2007)*, October 2007.

[48] X. Li and M. Orchard, "New edge directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, pp. 1521–1527, Oct. 2001.

[49] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, pp. 716–723, Dec 1974.

[50] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, vol. 11, pp. 416–431, Dec 1983.

[51] B. Atkins, *Classification based methods in optimal image interpolation*. PhD thesis, Purdue University, 1998.

[52] H. de Ridder, "Current issues and new techniques in visual quality assessment," in *IEEE International Image Processing*, vol. 1, pp. 869–872, September 1996.

[53] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. Prentice hall, 1999.

[54] D. Seidner, "Polyphase antialiasing in resampling of images," *IEEE Transactions on Image Processing*, vol. 14, pp. 1876–1889, Nov 2005.

[55] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Upper Saddle River, NJ 07458: Prentice Hall, 1993.

[56] S. M. Schweizer and J. M. F. Moura, "Modelling and detection in hyperspectral imagery," in *Proc. 1998 Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2273–2276, 1998.

[57] G. Rellier, X. Descombes, J. Zerubia, and F. Falzon, "A Gauss-Markov model for hyperspectral texture analysis of urban areas," in *Proceedings of the 16th International Conference on Pattern Recognition (ICPR 2002)*, vol. 1, pp. 692–695, 2002.

[58] J. P. Kerekes and J. E. Baum, "Spectral imaging system analytical model for subpixel object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, pp. 1088–1101, 2002.

[59] J. R. Schott, "Combining image derived spectra and physics based models for hyperspectral image exploitation," in *Applied Imagery Pattern Recognition Workshop*, pp. 15–24, 2000.

[60] D. Slater and G. Healey, "Physics-based model acquisition and identification in airborne spectral images," in *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, vol. 2, pp. 257–262, 2001.

[61] P. L. Vora, J. E. Farrell, J. D. Tietz, and D. H. Brainard, "Image capture: simulation of sensor responses from hyperspectral images," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 2692–2699, 2001.

[62] D. W. J. Stein, "Modelling variability in hyperspectral imagery using a stocastic compositional approach," in *International Geoscience and Remote Sensing Symposium (IGARSS 20001)*, vol. 5, pp. 2379–2381, 2001.

[63] R. R. Schultz and R. L. Stevenson, "Improved definition video frame enhancement," in *Proc. 1995 Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2169–2172, May 1995.

[64] L. Maloney, "Evaluation of linear models of surface spectral reflectance with small numbers of parameters," *Journal of the Optical Society of America A*, vol. 3, pp. 1673–1683, 1986.

[65] N. Keshava and J. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, pp. 44–57, Jan. 2002.

[66] A. Patti, M. Sezan, and A. M. Tekalp, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1064–1076, 1997.

[67] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1646–1658, 1997.

[68] P. Combettes, "Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections," *IEEE Transactions on Image Processing*, vol. 6, no. 4, pp. 493–506, 1997.

[69] P. Combettes, "Convex set theoretic image recovery with inexact projection algorithms," in *2001 IEEE International Conference on Image Processing (ICIP'01) Proceedings*, vol. 1, pp. 257–260, 2001.

[70] CALTECH Jet Propulsion Laboratory, "Aviris free data," Dec. 2007. http:/aviris.jpl.nasa.gov/html/aviris.freedata.html.

[71] Y. Altunbasak, R. M. Mersereua, and A. J. Patti, "A fast parametric motion estimation algorithm with illumination and lens distortion correction," *IEEE Transactions on Image Processing*, vol. 12, no. 4, pp. 395–408, 2003.

[72] J. C. Harsanyi and C. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, pp. 779–785, July 1994.

[73] J. C. Harsanyi, *Detection and classification of subpixel spectral signatures in hyperspectral image sequences*. PhD thesis, University of Maryland, 1993.

[74] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. V. der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Society for Industrial Mathematics, January 1, 1987.

[75] J. W. V. Miller, J. B. Farison, and Y. Shin, "Spatially invariant image sequences," *IEEE Transactions on Image Processing*, vol. 1, pp. 148–161, Apr. 1992.

[76] A. D. Stocker and A. Schaum, "Application of stochastic mixing models to hyperspectral detection problems," in *Proc. SPIE Algorithms for Multispectral and Hyperspectral Imagery III*, (Istanbul, Turkey), June 2000.

[77] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Dec. 2007. http://matrixcookbook.com/.

VITA

Toygar Akgun received the B.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 2001, and the M.S. degree in electrical engineering from Georgia Institute of Technology, Atlanta, in 2004. He will receive the Ph.D. degree in electrical engineering from Georgia Institute of Technology in 2008. His research interests include digital image/video processing, statistical signal processing and estimation theory.